MCR-70-419 Copy No. 29

Study of Potential Applications of Digital Techniques to the Apollo Unified S-Band Communications System

Phase 2 - Final Report

November 1970

prepared for

National Aeronautics and Space Administration
Manned Spacecraft Center

prepared by

MARTIN MARIETTA
DENVER DIVISION

CR-114921

MCR-70-419

Contract NAS9-9852

STUDY OF POTENTIAL APPLICATIONS OF DIGITAL TECHNIQUES TO APOLLO UNIFIED S-BAND COMMUNICATIONS SYSTEM

PHASE 2 - FINAL REPORT

November 1970

Approved

R B Blizard

Robert B. Blizard
Program Manager

MARTIN MARIETTA CORPORATION
DENVER DIVISION
Denver, Colorado 80201

FOREWORD

This report was prepared by the Martin Marietta Corporation in accordance with Contract NAS9-9852, "Study of Potential Applications of Digital Techniques to the Apollo Unified S-Band Communications System" for the Manned Spacecraft Center. This is the final report on all work performed on the second phase of this contract, during the period from February 23, 1970 to November 23, 1970. This report is submitted in accordance with Contract Schedule Article 7.

The NASA-MSC Technical Monitor was Mr. C. Kenneth Land whose direction and encouragement is hereby acknowledged. The Martin Marietta personnel contributing to this program were:

Dr. Robert B. Blizard	Program Manager
Ward B. Anthony	Software Studies
Clay C. Korgel	Coding Studies
James L. McKinney	Software Studies and Hardware Design
David L. Manion	Hardware Design and Image Processing
Gerald O. Olson	Hardware Design

CONTENTS

<u>Figure</u>

Table

I. INTRODUCTION

This contract was started in July 1969 as a seven-month study of digital data processing and error correcting code techniques and how they could be applied to the Apollo unified S-band communication system. The results of this first phase appear in a report published in February 1970.*

During the first phase it became apparent that certain parts of the study should be extended either to explore new and unforeseen possibilities or to get a more precise understanding of the original subjects. This report covers the work of the second phase.

Results of the first phase included selection of a zero order interpolator (ZOI) for the image compression and convolutional coding with Viterbi decoding for error correction. Performance of the image compression was demonstrated with single frames of digitized, compressed, and reconstructed pictures. Viterbi decoding was demonstrated by simulation on a general purpose computer. The next step, performed in the second phase, was to demonstrate the effects of the combination of compression and channel errors with Viterbi decoding. This is important because a single error can do a lot of damage to a compressed picture and the errors that come out of the Viterbi decoder tend to come in bursts. Chapter III of this report shows the results of a complete simulation of a picture that is digitized, compressed, encoded, transmitted over a noisy channel, decoded with the Viterbi algorithm, and reconstructed. The results verify the validity of combining error control with data compression as a means of reducing RF power without sacrificing picture quality.

Considerable hardware design, notably a Viterbi decoder to run at 7.5 Mb/s, was done in the first phase, but the objective was primarily to prove feasibility. In the second phase, as reported in Chapter II, we took a closer look at the flight hardware, particularly the ZOI compressor, and have made direct comparisons of the power, weight, and volume that are required for the present system and the digital system. The lower RF power needed for digital transmission permits the use of a solid state transmitter with large savings in volume, weight, and power.

**Potential Applications of Digital Techniques to Apollo Unified S-Band Communications System, Final Report*. MCR-70-34. Martin Marietta Corporation, Denver, Colorado, February 1970.

When testing Viterbi decoding by simulation, very long and expensive runs were needed to get good statistics, particularly in the region of low error rate. Therefore, included in the second phase is a computational program to determine a theoretical bound on error rate that is a good approximation to actual performance in the medium-to-high signal domain. This program, of which the description and results appear in Chapter VI, eliminates bad codes, chooses the best codes, and computes the union bound for constraint lengths from 4 to 9 and rate denominators 2, 3, and 4. For constraint lengths greater than 5, it is impractical to use paper and pencil to find good codes, and the tabulated computer results will be valuable to anyone working with Viterbi decoding.

During the study of Viterbi decoding, an iterative process for increasing performance was discovered. Subsequently this new technique developed into an independent decoding algorithm of great promise for space communication. Chapter IV describes the new algorithm and presents the analysis that predicts its performance in certain limiting cases and results of simulation on a general purpose computer. The results of computer simulation indicate that, for a rate denominator of 3, it outperforms sequential decoding. It is also self-starting and, unlike sequential decoding, it does not require periodic interruption of the data stream to insert a known sequence for recovery after a failure. Large constraint lengths are feasible (958 was used in the simulation), and therefore the system is resistant to fading. Chapter V describes a small breadboard decoder designed and built to demonstrate the feasibility of operation at 11 Mb/s.

II. IMPLEMENTATION STUDY

During the first phase of this contract the hardware for a digital S-band system was outlined with some parts designed to the logic element level. Since then we have made a more detailed and improved design of the zero order interpolator (ZOI) image compressor and the main elastic memory (or buffer) and its interfaces.

Comparisons are then made between a digital system and the present analog system for a simple replacement of equipment inside the Lunar Module (LM). An attractive alternative is to mount a solid-state transmitter directly on the erectable antenna to save the loss of RF power in the antenna cable. This arrangement will reduce the required antenna diameter by 60%.

A. DESCRIPTION OF DIGITAL HARDWARE

Figure II-1 is a block diagram of the digital system. There are three data sources in the space station that output data to be processed into a single PCM data train:

1) A TV camera with a commercial black-and-white format;
2) A 50.4 kb/s PCM telemetry train;
3) Two 3-kHz analog voice channels to be converted to digital.

Before performing the main stream multiplexing process which combines the four channels, a redundancy removal and run length encoding is performed on the TV video data. The multiplexed data are then encoded for error detection and correction.

As shown in Fig. II-2, 26 flat packs plus a crystal oscillator will be required for clocking functions, and power consumption is 260 mw. Main stream multiplexing will require an additional seven flat packs at 35 mw. Analog-to-digital conversion and two delta modulators require 30 dual in-line (DIP) packages at 680 mw. The convolutional encoder (V = 2 and K = 5) requires four flat packs and consumes 80 mw.

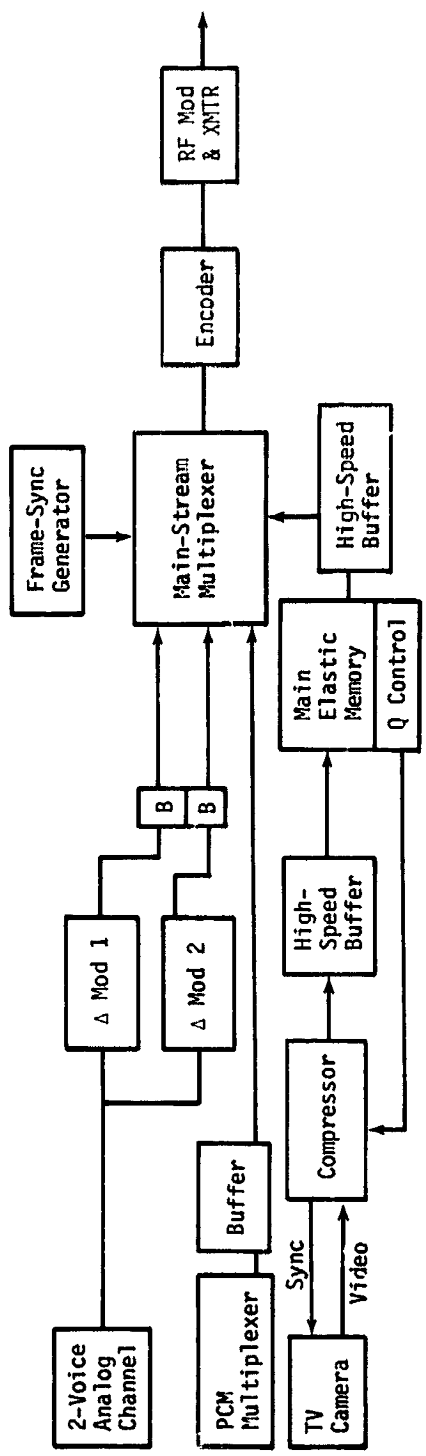


Fig. II-1 Transmitting End Block Diagram

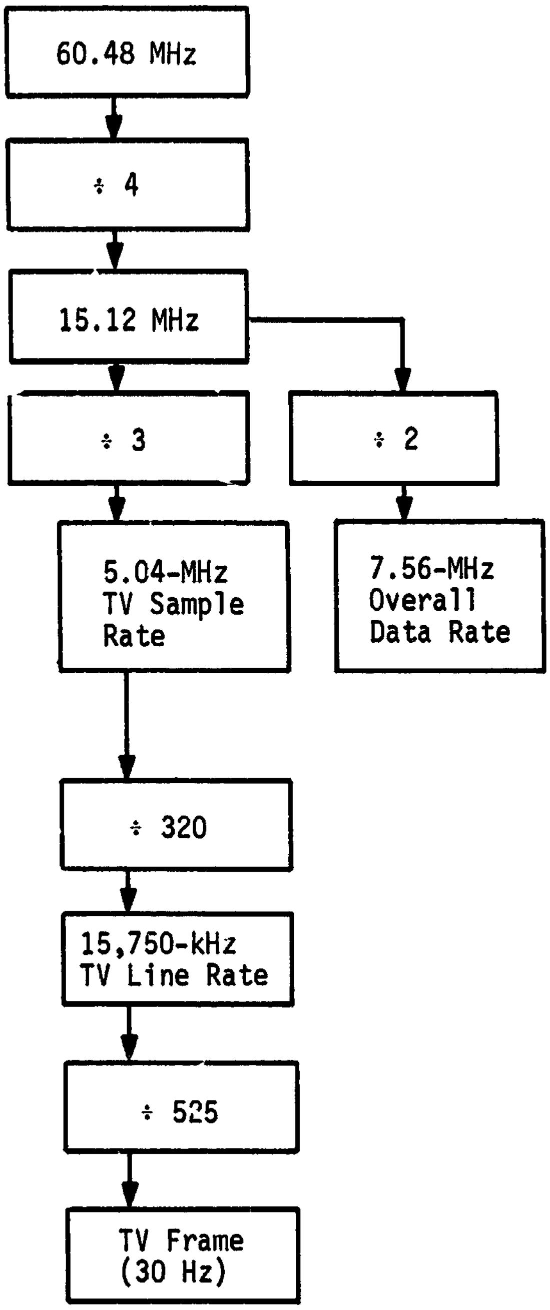


Fig. II-2 Typical Station Clock Relationships

Figure II-3 is a block diagram of the ZOI compressor using standard TTL logic elements and MSI.

The major hardware problem in using this type of data compression algorithm is working with a variable word length at a variable rate at the memory input and a fixed word length at a fixed rate at the output. Speed further complicates the problem. For instance, the hardware under consideration can generate 8 bits of data every 200 ns -- a 5-bit gray level and a 3-bit run length code -- as a worst case when processing successive nonredundant data samples. The memory, however can be accessed for read or write only at a 1 MHz rate, thereby setting up the requirements for a 5:1 speed reduction at the memory interface.

An efficient method of realizing the 5:1 speed reduction is to make the memory input word equal five words from the data compressor hardware -- five words each made up of a 5-bit gray level plus a 3-bit run length code -- for a memory input word of 40 bits minimum. This technique serves to solve a second problem equally important -- minimizing address generation. If an MOS-RAM is used, the entire 40-bit (or more) input word can have a single major address, and any portion of the word may be accessed within the address, by use of the "chip select" function. Each chip accessed dissipates 0.5 w, and the worst-case condition of successive nonredundant 8-bit data words at a 200 ns sample time requires 20 w for memory input. However, this is the maximum power, and the average power is 3.6 w corresponding to the average bit rate, 7.1 Mb/s. The input word size can be increased to any desired length without changing maximum power required, as only 40 chips are being accessed during a microsecond. The memory size is sufficient to assure data will be available during retrace, interfield, and end-of-frame dead times when data are not being generated, as well as providing an effective buffer to smooth the data rate over dull and active portions of the picture.

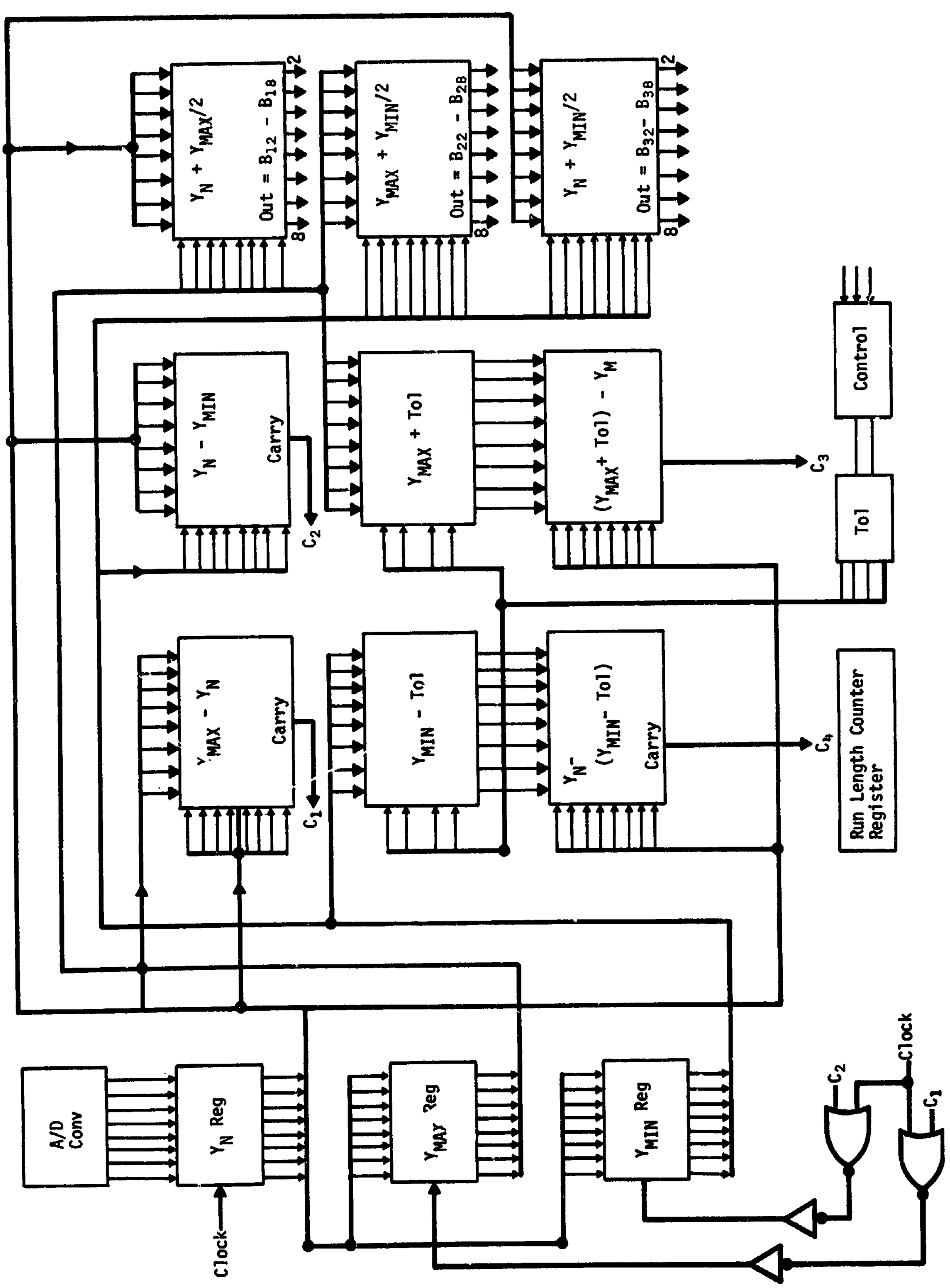


Fig. II-3 ZOI Block Diagram

A method of presenting the variable length word from the data compressor (5-bit gray level plus 3- to 7-bit run length code) to the memory as a fixed length word is shown in Fig. II-4 and II-5. Figure II-4 is the block diagram and Fig. II-5 is a more detailed logic diagram. A 16-bit register (to handle successive maximum length 12-bit data words) temporarily stores the variable length data word from the compressor in the proper bit positions depending on the length of the preceding data word. Bits 1 thru 8 of the 16-bit intermediate buffer are transferred to the successive groups of 8 bits of the memory input word buffer where they are read into the main memory during the following microsecond. In the meantime, the data in bits 9 to 16 of the intermediate buffer are transferred into its corresponding leading bits (i.e., bit 9 to 1, 10 to 2, etc) and simultaneously the next data word from the compressor is pointed to the correct positions in the 16-bit buffer. In other words, the effect simulates a set of 12-bit registers, each offset from the preceding by one bit, and each selected to match the solution of a positioning equation to address positions in the 16-bit buffer, the register number being the position of the first bit in the current data word.

None of the buffers and registers need be cleared and each bit requires only a single input. It is part of the operating sequence to overlay invalid data, and data need be correct only while being read into the main memory.

Figure II-6 and Table II-1 will make this clearer.

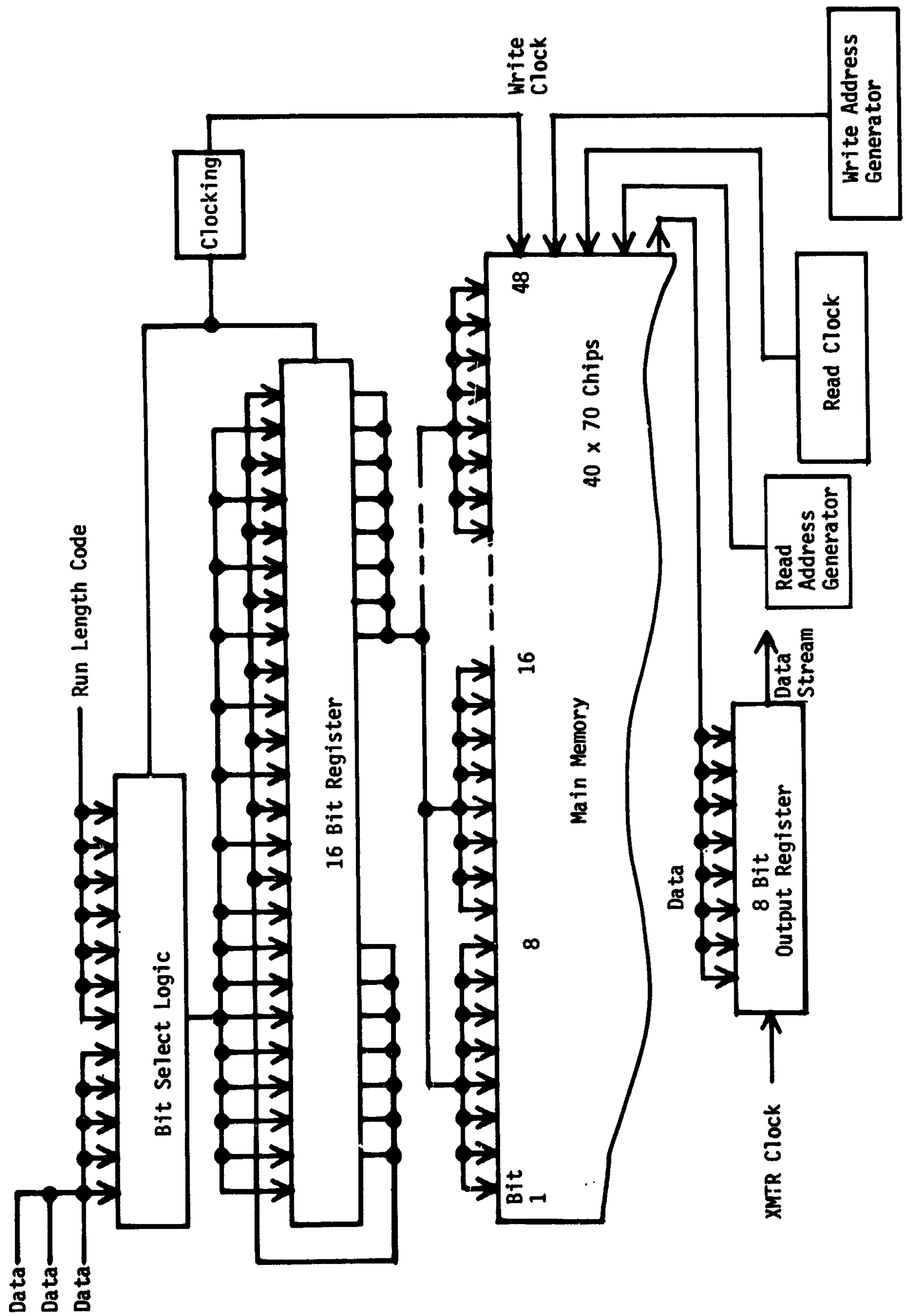


Fig. II-4 Block Diagram of Memory Input Buffer

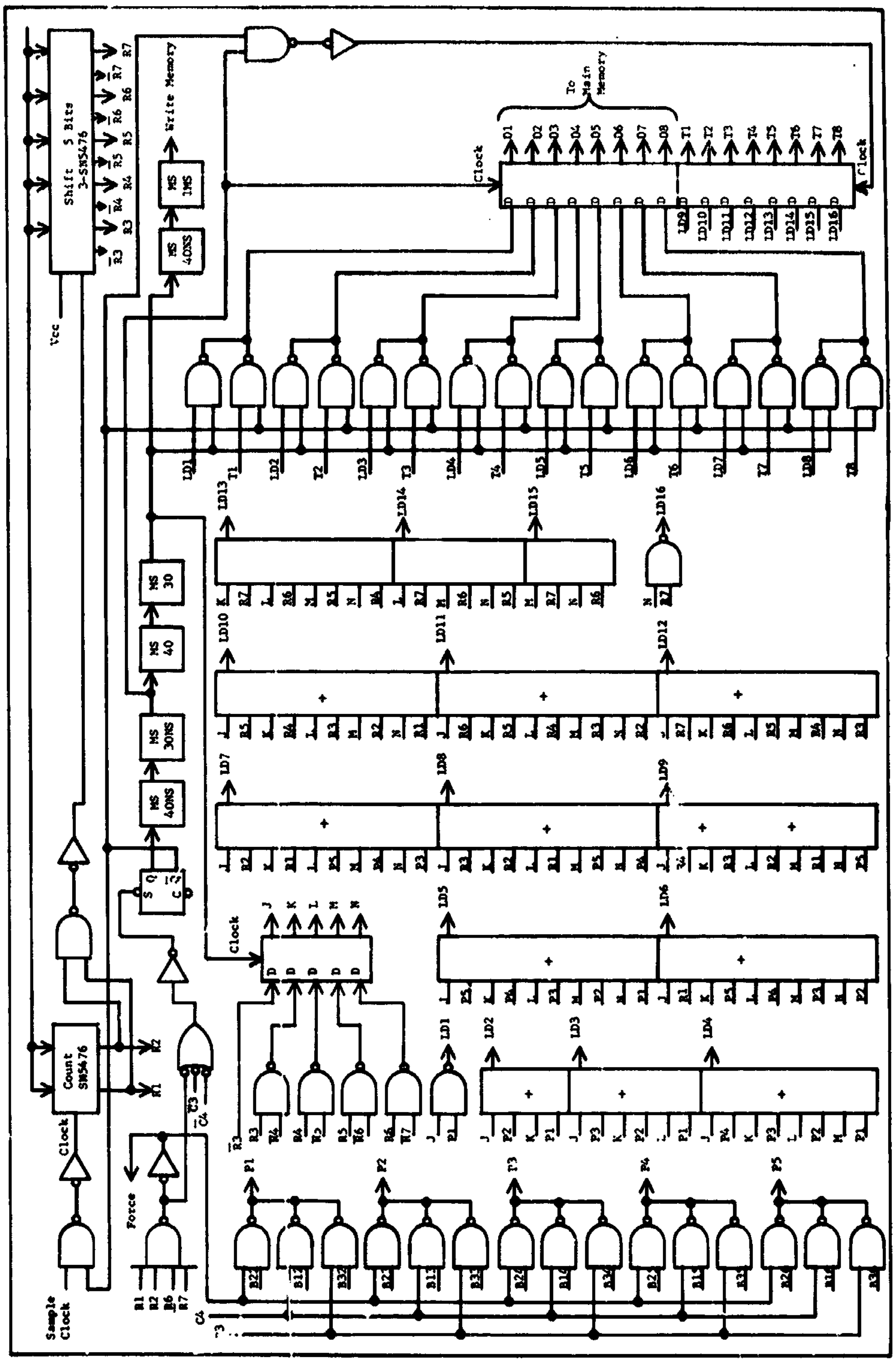


Fig. II-5 Logic Diagram for Memory Input Buffer

	Position
16-Bit Buffer	1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16
Register 0	1 2 3 4 5 6 7 8 9 10 11 12 — Data Word Bits
1	1 2 3 4 5 6 7 8 9 10 11 12
2	1 2 3 4 5 6 7 8 9 10 11 12
3	1 2 3 4 5 6 7 8 9 10 11 12
4	1 2 3 4 5 6 7 8 9 10 11 12

(Next register number) = (previous register number) + (current data word length) - (number of bits transferred to memory input buffer)

Fig. II-6 Schematic Arrangements of the 16-Bit Buffer and the 12-Bit Registers

Table II-1 Rules for Number of Bits To Be Transferred and for the Next Register Number

Previous Register Number	Current Data Word Length	Bits Transferred	Next Register Number
0	8	8	0
	9	8	1
	10	8	2
	11	8	3
	12	8	4
4	8	8	4
	9	12	1
	10	12	2
	11	12	3
	12	16	0

Note: If any register number greater than 4 is indicated, an additional 4 or 8 bits are read out of the 16 bit register before repeating the operating sequence.

The sequence of events is:

1) Read data word from compressor into the 16-bit buffer specified by the register number. This is in reality a beginning position address to the 16-bit buffer;

2) Transfer data from the leading 4 or 8 bits of the 16-bit buffer to the proper memory input buffer positions (see step 4);

3) Move the data in the remainder of the 16-bit buffer to the leading bit positions;

4) Determine the beginning bit position (register number) for the next data word. If this is greater than 4, loop back to step 2 and continue; otherwise go to step 5;

5) Start the 1 μsec read-in to the main memory of the memory input buffer bits just loaded (8, 12, or 16). Return to step 1.

Consider the following example in which data words 4, 5, and 6 (DW4, DW5, and DW6) are handled. Let the lengths of these words be 9, 8, and 11 bits respectively.

Assume register 3 has been selected for DW4. The sequence of operations is then as follows:

1) DW4 is inserted in positions 4-12 by register 3 addressing;

2) Data in positions 1-8 are transferred to memory input buffer bits 25-32 (write No. 1 filled bits 1-8, write No. 2 bits 9-16, and write No. 3 filled bits 17-24);

3) Data in remaining positions are inserted into leading positions;

4) Previous beginning position (3) + DW4 length (9)-8 = register 4 addresses selected for next word;

5) DW5 is inserted into positions 5-12 overlaying invalid data in those positions with new valid data. Position 1-4 contains remainder of data from DW4 and positions 5-8 contain the first 4 bits of data in DW5. Positions 9-12 hold the last 4 bits of data in DW5;

6) Data in positions 1-8 is transferred to memory input buffer bits 33-46;

7) Operation 3 is repeated;

8) Operation 4 is repeated - register 4 again being selected for next word;

9) DW6 is inserted into positions 5-15 by register 4 selection. Positions 1-4 contain remainder of DW5; positions 5-8 have first 4 bits of DW6;

10) Data in positions 1-8 are transferred to memory input buffer bits 1-8;

11) Bits 1-4 are transferred to bits 9-12 of memory input buffer. Register 3 now indicated for next data word and system ready to continue.

The compressor hardware in addition to memory requires about 20 packages and 2 w. The logic and functions peripheral to the main memory require 3.5 w and 45 packages. The main memory is composed of 380 packages in a 40 x 7 (package) format which gives 9k 8-bit words of data. The average power required is 3.6 w at input and 3.6 w at output, plus 3.5 w standby. Changing the size of the memory or the format (within the restrictions of a 40-bit input word minimum and 8 bits out fixed) will change only the standby power at 50 μw/bit.

There is essentially no difference to the package count in implementing the peripheral logic for the memory system with gates, flip-flops, and/or invert packages, or a mix.

If no bit packing is done at the input, but rather a fixed length word (5 bits for pixel value and five bits for a run length count) an additional 57 packages are required in the memory for the same capability, and memory power requirements rise to 5 w at the output. The decoding and reformatting at the output, into a pixel-run length code word is simpler and will require, for hardware peripheral to the memory, only 1.5 w and about 15 packages -- however, this implementation means a net increase of 27 packages (because of additional memory packages) and 4.5 w more than working with the variable word at the input.

Table II-2 shows the power, weight, and volume needed for the digital circuitry.

Table II-2 Power, Weight, and Volume of Digital Components

Component	Power (w)
Delta Modulators (2)	0.280
Main Stream Multiplexing	0.035
PCM-Voice Buffer	0.160
Convolutional Encoder	0.080
Clocking	0.260
A/D Conversion	0.400
ZOI Compressor	2.000
Memory Interface	3.500
Main Memory	10.700
Output Buffer	0.320
Total Power	17.7
Total Weight (lb)	2.2
Total Volume (cu in.)	36.0

B. TRANSMITTERS

As discussed in the final report to the first phase of the contract, the digital system using image compression and a K = 5, V = 2 convolutional code will require about 5 dB less effective radiated power than the analog system. Many options are available to exploit this advantage, including greater range, smaller antennas, lighter and less power-consuming transmitters, and combinations of these. As substitutes for the transmitter, three possibilities will be considered: an amplitron with lower power, a flight-qualified off-the-shelf solid-state transmitter, and a state-of-the-art solid-state transmitter with higher efficiency.

Proven amplitrons are available with a wide range of power ratings. For all of them the efficiency is about the same, and therefore the input power can be scaled according to the reduction in radiated power permitted by the higher efficiency of the digital system. Table II-2, therefore shows the same weight and volume as the present power amplifier but a power consumption that is 5 dB lower.

Off-the-shelf S-band transmitters may be typified by the Motorola MTT-501. This model is normally furnished with an FM modulator, but PM would not change the main parameters. It is less efficient than the TWT amplifier, and the power saving is only about 22%, but the savings in weight and volume are large.

A solid-state S-band amplifier with high efficiency is described by G. J. Gilbert.* Its power output is 10 w at 2.2 to 2.3 GHz, gain is 23 dB, and DC to RF efficiency is 30%. Thus, the output power is reduced to 6 w, and size and weight are assumed the same as for the off-the-shelf transmitter (Table II-3).

Table II-3 Power, Weight, and Volume for S-Band Transmitters

Transmitter	RF Output (w)	Input Power (w)	Weight (lb)	Volume (cu in.)
Present Apollo Power Amplifier	18.6	73.5	16.8	432
Amplitron with Lower Power	5.8	24.0	16.8	432
Off-the-Shelf Solid-State Transmitter including Modulator	6.0	57.0	2.3	32
State-of-Art Solid-State Power Amplifier	6.0	20.0	2.3	32

*George J. Gilbert: *Microwave Transistors: A Look to the Future Microwaves*, July 1970.

C. COMPARISON

An exact comparison between digital and analog systems is difficult because of their fundamental differences. The data handling is organized differently, and the individual pieces of equipment in one system do not have exact counterparts in the other. The digital equipment listed in Table II-2 performs almost all the functions of the LM signal processor assembly. For the purposes of this comparison, therefore, it is assumed that the digital system with its transmitter will replace the signal processor and the power amplifier. In Table II-4 the power, weight, and volume are given for the combinations described above, and for the digital system the savings are shown in parentheses below the actual values.

Table II-4 Comparison of Power, Weight, and Volume of Analog and Digital Systems

System	Power (w)	Weight (lb)	Volume (cu in.)
Present Signal Processor	16.5	10.0	364
Present Power Amplifier	73.5	16.8	432
Total for Present Analog System	90.0	26.8	796
Digital system with:			
Low Power Amplitron	41.7 (-53%)	19.0 (-29%)	468 (-41%)
Off-the-Shelf Solid-State Transmitter	74.7 (-17%)	4.5 (-83%)	68 (-94%)
State-of-Art Solid-State Transmitter	37.7 (-58%)	4.5 (-83%)	68 (-94%)

The small size and weight of the solid-state transmitter makes it possible to achieve further hardware reduction when the erectable LM antenna is used. With the present arrangement, RF power is fed to the antenna through a cable, and circuit loss amounts to 7.7 dB. This loss could be largely avoided by mounting the transmitter directly on the antenna, and the antenna could then be much smaller. This in turn would decrease the pointing loss and increase the pointing tolerance. Table II-5 shows the balance for the two systems. The antenna-mounted transmitter could be interchangeable with the one in the LM and they could serve as mutual spares. Advantages would include much less storage space for the erectable antenna, less operational time for erection, and easier pointing.

Table II-5 Comparison of Present FM System with Digital System Using Antenna-Mounted Solid-State Transmitter

Item	Present System	Solid-State System
Transmitter Power	18.6 w 12.7 dBw	7.0 w 8.4 dBw
Circuit Loss	-7.7 dB	-1.0 dB
Pointing Loss	-2.0 dB	-0.7 dB
Coding and Compression Gain		5.0 dB
Antenna Gain	Approx 10 ft dia 34.0 dB	Approx 4 ft dia 26.0 dB
Total	37.0 dB	37.7 dB

III. IMAGE PROCESSING

During the first phase of this contract image compression was demonstrated using zero-and first-order predictors and interpolaters. The zero-order interpolater (ZOI) was chosen as most suitable for this application because of its good compression performance and its modest hardware requirement. The transmitted information is encoded into words of variable length. The first five bits (in the selected version) indicate one of 32 possible gray levels and the remaining bits indicate how many successive picture elements or "pixels" are to be given that gray level. This number is called the run length, and the code for the run length is of variable length, three bits for 1 through 4, four bits for 5 through 8, and so forth up to length 20. The last bit of each word is a 0, called a comma, to indicate to the reconstructing system that the word is terminated.

With such a system, unless precautions are taken, a single bit in error can upset a large part of the picture that is being transmitted. An error in the gray level code will persist for a run length. An error in the run length code will displace all subsequent pixels. An error in a comma, or an error that inserts a comma where none should be, completely upsets reconstruction of the line from that point on. To limit propagation of the effects of errors, a special code is used at the end of each line that cannot be mistaken for any segment of ordinary code words. Therefore no matter how badly the timing of a line is upset, the reconstruction system will recover on the first end-of-line code that it receives correctly.

To verify this operation and to permit a subjective evaluation of the effects of noise with the combination of Viterbi decoding and image compression, the pictures shown in Fig. III-1 thru III-5 were made. Figure III-1 shows the original digitized picture with 64 gray levels and 512 x 512 pixels; only 32 levels of gray are used in the compressed pictures because of the dither. Figure III-2 shows the same picture with half the pixels ignored (in a checkerboard pattern) and the remaining pixels transmitted with an average of 1.8 bits per remaining pixel. The three other pictures show the effect of the kind of noise that would be seen in a channel using a convolutional code with constraint length 5 and the rate ½ at signal-to-noise ratios per bit $\left(E_b/N_o\right)$ of 2.5, 3.0, and 4.0 dB

It is obvious that the end-of-line codes are effective in getting reconstruction back on the track. The frequency of these special codes is a parameter of the system that may be adjusted for optimum performance. If middle-of-line codes were also used, the error streaks would only be half as long, but signal energy would be diverted from the main data and there would be more errors. Similarly, signal energy could be saved by inserting the special code words only after every other line, thus making half the streaks extend into the next line. The choice of this parameter is not critical, but the optimum seems to be near 1 per line, and this is a convenient arrangement.

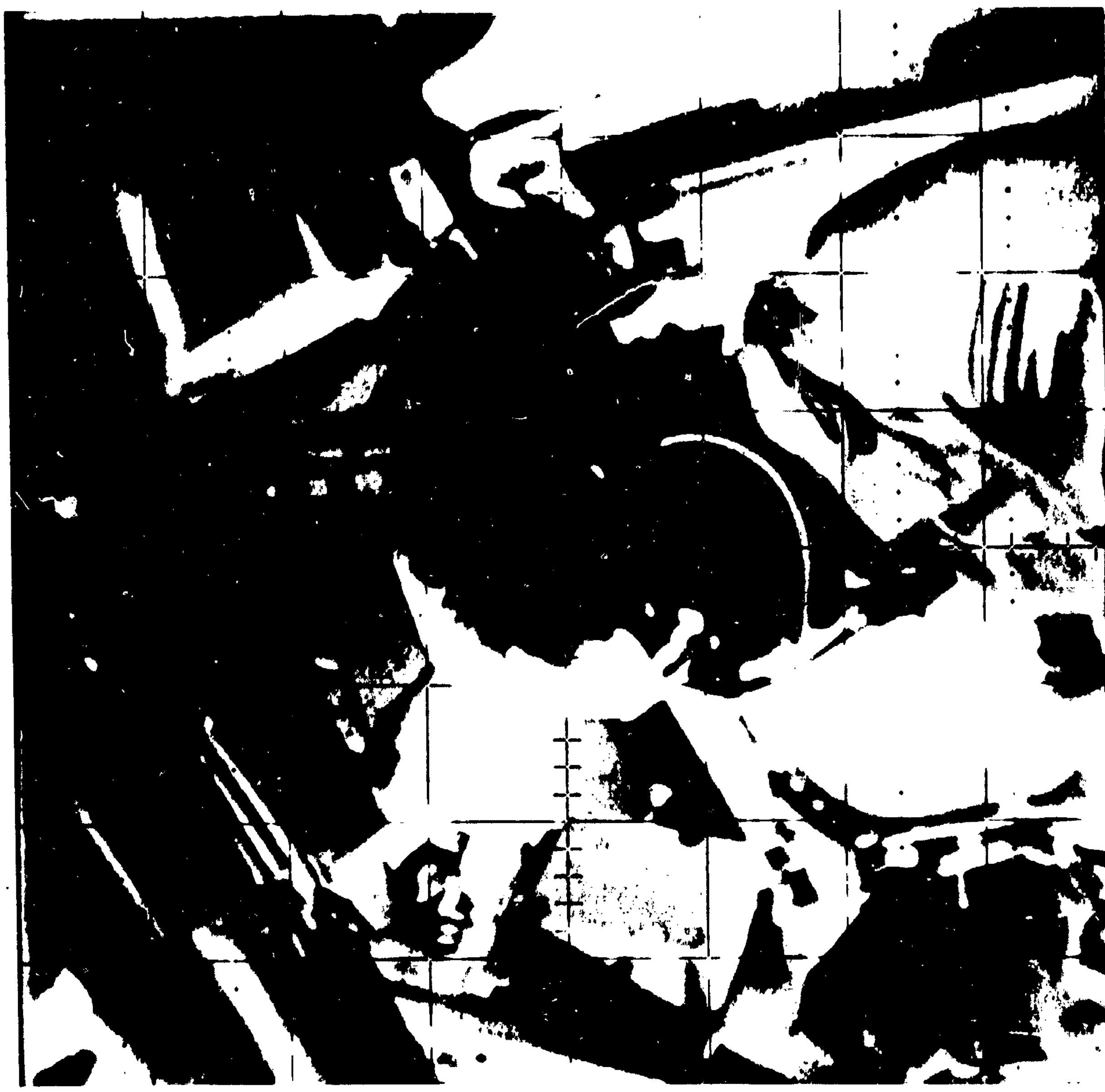


Fig. III-1 Original Digitized Picture

Fig. III-2 Compressed Picture without Noise

Fig. III-3 Compressed Picture, $E_b/N_o = 2.5$ dB

Fig. III-4 Compressed Picture, $E_b/N_o = 3.0$ dB

Fig. III-5 Compressed Picture, E_b/N_o = 4.0 dB

IV. NEW CODING ALGORITHM

This chapter describes a new algorithm for decoding error-correcting codes. It was originally conceived during the performance of this contract as a superstructure to be applied to Viterbi decoding to permit the use of a large constraint length K without an exponential increase in computation. It was also analyzed, independently of Viterbi decoding, as a technique that could be applied to an erasure channel at rates up to channel capacity. These results were described in the final report for the first phase of this contract.

It was later realized that it was much simpler to use the new algorithm directly without Viterbi decoding as an intermediate step. Analysis showed that, for large K and large rate denominator V, data rates can approach channel capacity with arbitrarily small probability of error. The amount of decoding computation is proportional to the minimum code word weight, but is independent of constraint length. Decoding is self starting, and a true convolutional code can be used without periodic resetting. The analysis and properties of the new algorithm were described in an oral presentation at NASA/MSC on April 2, 1970.

Since that time we have learned how to use the algorithm and have obtained simulation results for V = 2 and 3. Relative to other available decoding methods, the new algorithm has good performance with V = 2 and superior performance with V = 3. An obvious next step, if funds become available, is to try V = 4 and 5. We believe that for channels that have enough band for V = 3, the new algorithm can achieve better performance at a reasonable cost than any other method.

A. ALGORITHM

Consider a message consisting of a sequence of 1's and 0's called *bits*. These are encoded into a larger number of 1's and 0's, called channel *symbols*, each of which is formed as the mod-2 sum of one or more of the message bits. If only a single bit is included in a sum, then the symbol is obviously equal to the bit, and these symbols are called *systematic symbols*. The other symbols are called *check symbols*. If each bit is used to form at least one systematic symbol, the message can be read directly from these symbols, and the code is said to be a *systematic code*. The new decoding algorithm uses a systematic code.

The symbols are transmitted over a noisy channel and are detected at a receiver with some probability of error. Some receivers make a *hard decision*, putting out a stream of 1's and 0's that are the best guesses (independent of the code structure) of the individual symbols that were transmitted. Others make a *soft decision* and also indicate the reliability of each guess according to the magnitude of the received signal or by some other criterion. In either case the raw material that the decoder works with is the array of receiver outputs, or symbol estimates, that indicate the probability, for each symbol, that it is a 1 or 0.

Although the algorithm can be used with either block or convolutional codes, a convolutional code is assumed for the description of the decoding process.

As the first step in the decoding algorithm, the systematic symbol estimates are copied to form the *zero order message estimates*. These message estimates and the check symbols are then used to obtain an improved set of *first order* message estimates from which further improvements will be obtained by iteration until the message estimates converge or until the available time is used up. To understand how this process is done, suppose first that all but one of the bits have been decoded with great confidence. To decode the remaining uncertain bit, all the check symbol estimates derived from this bit can be used. It is only necessary to compute for each symbol estimate, the parity of the known bits that contributed to it and to reverse it if the parity is odd. In this way all the symbol estimates derived from this last uncertain bit can be used to decode it, and the chances for success are high.

The above case, with one or a very few, uncertain bits, applies to the last stages of decoding. Before this state is reached, the decoder must work with message estimates that are largely unreliable, and the probabilities of error must be used in calculating the next set of message estimates.

At this point some notation is required. M(I) is the I'th message bit, either a 1 or a 0. E(L,I) is the message estimate for the I'th bit after L decoding iterations. Each E is a number representing the decoder's best guess, at each iteration, of the logarithm of the ratio of probabilities of 0 and 1. Thus,

$$E(L,I) \simeq \ln \frac{P(M(I) = 0 \mid L)}{P(M(I) = 1 \mid L)}, \qquad [IV\text{-}1]$$

where the condition L is meant to indicate "all information used by the decoder through the L'th iteration."

The channel symbols are designated S(I,J). I is the bit index introduced previously, and J is the index of the mod-2 adder in the code generation. J takes the values from 1 to V where V is the rate denominator -- that is, the rate is 1/V. S(I,1) is the systematic symbol for the I'th bit.

Q(I,J) is the estimate of S(I,J) based only on the received signal. It is not recomputed as decoding progresses. Like E, it is the natural logarithm of the probability ratio:

$$Q = \ln \frac{P(S = 0 \mid R)}{P(S = 1 \mid R)}, \qquad \text{[IV-2]}$$

where R signifies the received signal corresponding to the symbol. For a binary antipodal signal and a memoryless channel with Gaussian noise, Q is proportional to the signal voltage.

For both E and Q, large positive values indicate high probability of 0, large negative values high probability of 1, and values near zero indicate uncertainty.

The first step in decoding is to set the zero-order message estimates equal to estimates of the systematic symbols:

$$E(0,I) = Q(I,1). \qquad \text{[IV-3]}$$

The check symbols estimates are used on the next iteration, but they must be combined with the zero-order estimates E(0,I), as will be explained. For the sake of concreteness, suppose that the second mod-2 adder in the encoder is connected to stages 1, 3, and 6 of the shift register so that

$$S(I,2) = M(I) \oplus M(I-2) \oplus M(I-5), \qquad \text{[IV-4]}$$

where $\oplus$ indicates addition mod-2. Thus Q(I,2) the estimate of S(I,2), can be used to help in decoding M(I), M(I-2), and M(I-5). Another way to state the same thing is to say that Q(I,2), Q(I+2,2), and Q(I+5,2) can all be used to help decode M(I).

To use Q(I,2) to decode M(I), its sign is changed if one and only one (an odd number, in the more general case) of E(0,I-2) and E(0,I-5) is negative. This is an attempt to solve Eq [IV-4], for M(I) using the best information available at the receiver. If we were sure of M(0,I-2) and M(0,I-5), we would add Q(I,2) to Q(I,0) and any other appropriate Q's to get the next estimate E(1,I). Because of the uncertainty, however, a smaller absolute value than that of Q(I,2) should be used, and this value should be the natural logarithm of the probability ratio for the combined guesses for S(I,2), and M(O,I-2), and M(O,I-5).

To calculate this quantity, consider a number of independent binary random variables with probability p_i of being 1 and q_i of being 0. As is expected, $p_i + q_i = 1$. Define the probability ratio $u_i = q_i/p_i$ and the probability difference $b_i = q_i - p_i$. We want to calculate the probability ratio U for the mod-2 sum of the original variables. It is easily shown that the probability difference B for the mod-2 sum is

$$B = \prod_i b_i \qquad [IV-5]$$

(Write the b_i in terms of q_i and p_i and expand. Positive terms, those with even numbers of p's, add to give the probability of 0 for the sum, and the negative terms add to the probability of 1.)

The b_i are obtained from the u_i as follows:

$$b_i = q_i - p_i = \frac{q_i - p_i}{q_i + p_i} = \frac{u_i - 1}{u_i + 1} \qquad [IV-6]$$

Since our message and symbol estimates are in logarithmic form it is convenient to use

$$\ln b_i = \ln \frac{\exp\left(\ln u_i\right) - 1}{\exp\left(\ln u_i\right) + 1} = \ln \tanh \frac{\ln u_i}{2} . \qquad [IV-7]$$

To obtain ln U from ln B the following steps may be taken.

$$U = \frac{1 + B}{1 - B}$$

$$-\ln U = \ln \frac{1 - \exp(\ln B)}{1 + \exp(\ln B)} = \ln \left(\tanh \frac{-\ln B}{2}\right). \qquad [IV-8]$$

Defining a function

$$F(x) = -\ln \tanh(x/2) \qquad [IV-9]$$

permits some simplification.

$$\ln B = \sum \ln b_i \qquad [IV-10]$$

$$= \sum -F\left(\ln u_i\right) \qquad [IV-11]$$

$$\ln U = F(-\ln B) \qquad [IV-12]$$

$$= F\left[\sum F\left(\ln u_i\right)\right] . \qquad [IV-13]$$

For decoding it is only necessary to use Eq [IV-13] with the message and symbol estimates:

$$\begin{aligned} E(1,I) = {} & Q(I,1) \\ & + AF \quad [F(Q(I,2)) + F(E(0,I-2)) + F(E(0,I-5))] \\ & + BF \quad [F(Q(I+2,2)) + F(E(0,I+2)) + F(E(0,I-3))] \qquad [IV-14] \\ & + CF \quad [F(Q(I+5,2,)) + F(E(0,I+5)) + F(E(0,I+3))] \end{aligned}$$

In Eq [IV-14], A, B, and C are equal to -1 or +1 according to the signs of the Q and E's in each term.

On the next iteration, E(2,I) is calculated using E(1,I) in place of E(0,I) in Eq [IV-14].

B. EXPECTED PERFORMANCE

In this section decoding behavior is analyzed in the limiting case of a large code with many channel symbols per message bit. The result is that, at all rates below channel capacity and with proper code design, decoding converges, and the probability of error can be made arbitrarily small.

1. Idealized Channel

Consider a channel that accepts a memoryless symmetric binary input (1 or 0) and has M output states x_i, i = 1 to M. For each output state there is a certain probability that 0 was transmitted:

$$P\left[0 \middle| x_i\right] = \frac{P\left[x_i \middle| 0\right] P\,[0]}{P\left[x_i \middle| 0\right] + P\left[x_i \middle| 1\right]} \qquad [IV-15]$$

It will be convenient to use the probability difference, introduced earlier in the description of the algorithm. For each receiver state x_i, define

$$b_i = P\left[0 \middle| x_i\right] - P\left[1 \middle| x_i\right] \qquad [IV-16]$$

It is easily verified (see subsection 2, following) that the channel capacity in bits per symbol is

$$C = \frac{1}{2 \ln 2} \sum_i b_i^2 P\left[x_i\right] \qquad [IV-17]$$

in the limit as all b_i approach zero.

Assume a large block code constructed as follows. There are K message bits in each block and these are encoded into KV channel symbols. Both K and V are assumed to be large.

A fraction A_1 of the symbols correspond directly to the K message bits. Thus on the average there are VA_1 systematic channel symbols per message bit. If each bit controls at least one of these symbols, the code is systematic, but this is not necessary to prove that channel capacity can be approached with arbitraty probability of error.

A second fraction A_2 of the symbols are formed as mod-2 sums of pairs of message bits, and in general a fraction A_i are formed as mod-2 sums of i message bits. The sum of all the A_i is, of course, 1. The code should be selected so that each message bit controls, by itself and with other bits, approximately the same number of symbols. Also, of the symbols controlled by a given bit, no more than a small fraction should also be controlled by any other given bit. More generally, the grouping of bits should be arranged so that there is no small subset that shares control of too many symbols. The purpose of these restrictions is to assure that in the decoding process the probabilities of error for the individual message bits will be statistically independent.

For simplicity assume that the message, and hence the transmitted code word, is all 0's. The following paragraphs discuss averages of message estimates, symbol estimates, etc, and it is convenient to omit the bit index from the notation. Thus $\overline{E}(L)$ is the average message estimate after L iterations of using the check symbols.

For each message bit there are, as stated in the previous paragraph VA_1 systematic symbols. Therefore,

$$\overline{E}(0) = VA_1\overline{Q} \qquad [IV-17]$$

where $\overline{Q}$ is the mean symbol estimate. For the small signal limit, Q can be simply expressed in terms of b:

$$Q = \ln \frac{1+b}{1-b} = 2b \qquad [IV-18]$$

in the limit as b approaches 0.

Equation [IV-17] now becomes

$$\overline{E}(0)/2 = VA_1\overline{b}. \qquad [IV-19]$$

When the check symbols are used the first time as indicated previously in Eq [IV-14],

$$\overline{E}(1)/2 = V\overline{b} \sum_i iA_i \left[\overline{B}(0)\right]^{i-1} \qquad [IV-20]$$

In this equation, the systematic estimates appear as usual, and the VA_i check symbols that are formed from i bits are each used i times. $\overline{B}(0)$ is the average probability difference for the zero-order message estimate. Because the b's and B's are statistically independent, the mean of their products are the products of their means. Each of the terms in Eq [IV-20] is the logarithm of the probability ratio for a statistically independent event affected by the transmitted message, and therefore $\overline{E}(1)$ is the mean of the logarithms of the probability ratios for the message bits after this iteration.

For subsequent iterations, Eq [IV-20] is modified by substituting the general iteration indices L and L + 1 for 0 and 1. It is now necessary, however, to use the assumption of large V and very diffuse code to assure that the B's are statistically independent. With a finite code, the message estimates, and thus the B's that are used to decode a given bit were all influenced by that bit in the previous iteration. This is of practical as well as theoretical importance.

E/2 is a sum of a large number of statistically independent terms, and we can invoke the central limit theorem to assert that E(L) is normally distributed with mean $\overline{E}$. Furthermore each of the individual terms is a product of probability differences and is therefore also a probability difference. For any probability difference b,

$$\overline{b} = \overline{b^2} \qquad [IV-21]$$

as can be seen from the following. Let the probability density function for b be p(b). Then

$$b = \frac{p(b) - p(-b)}{p(b) + p(-b)} \qquad [IV-22]$$

$$\overline{b} = \int_{-1}^{1} bp(b)db = \int_{0}^{1} b\ [p(b) - p(-b)]\ db$$

$$= \int_{0}^{1} b^2\ [p(b) + p(-b)]\ db = \overline{b^2} \qquad [IV-23]$$

The variance of b is $\overline{b^2} - \overline{b}^2 = \overline{b} - \overline{b}^2$, and for small b the variance is simply $\overline{b}$. Thus the variance of E/2 is the sum of the variances of the individual terms or $\overline{E}/2$. The probability density function for E(L) can now be written:

$$p[(E(L)] = \frac{1}{\sqrt{4\pi \overline{E}(L)}} \exp - \frac{[E(L) - \overline{E}(L)]^2}{4\overline{E}(L)} \cdot \qquad [IV-24]$$

Since $B(L) = \tanh \frac{E(L)}{2}$, it is now possible to calculate $\overline{B}(L)$ as a function of $\overline{E}(L)$. The resulting integral has been computed and is shown in Fig. IV-1. Also shown is a curve, for some set of A_i and some signal-to-noise ratio, of $\overline{E}(L+1)$ as a function of $\overline{B}(L)$ as in Eq [IV-20].

As long as this curve is to the right of the curve of $\overline{B}$ as a function of $\overline{E}$, each pass gives larger values of $\overline{E}$ and $\overline{B}$. This is indicated by the arrows between the curves in Fig. IV-1.

The trick in designing the code to work near channel capacity is to choose the A_i to make the curve for $\overline{E}$ match that of $\overline{B}$ as closely as possible while staying to the right of it until a very high value of $\overline{E}$ is reached.

When the curves match, the areas above them must be equal, and

$$\int_0^1 \overline{E}(L+1)d\overline{B} = \int_0^\infty 1 - \overline{B}(L)d\overline{E} \qquad [IV-25]$$

The integral on the left can be evaluated immediately from Eq [IV-20]:

$$\int_0^1 \overline{E}(\overline{B})d\overline{B} = 2V\overline{b}\sum A_i = 2V\overline{b} \qquad [IV-26]$$

The other integral has been evaluated numerically and the result is 2.772568, which is within 2.0×10^{-5} of 4 ln 2. If, as seems likely, the true value is 4 ln 2, then from Eq [IV-21]

$$V\overline{b}^2 = 2 \ln 2. \qquad [IV-27]$$

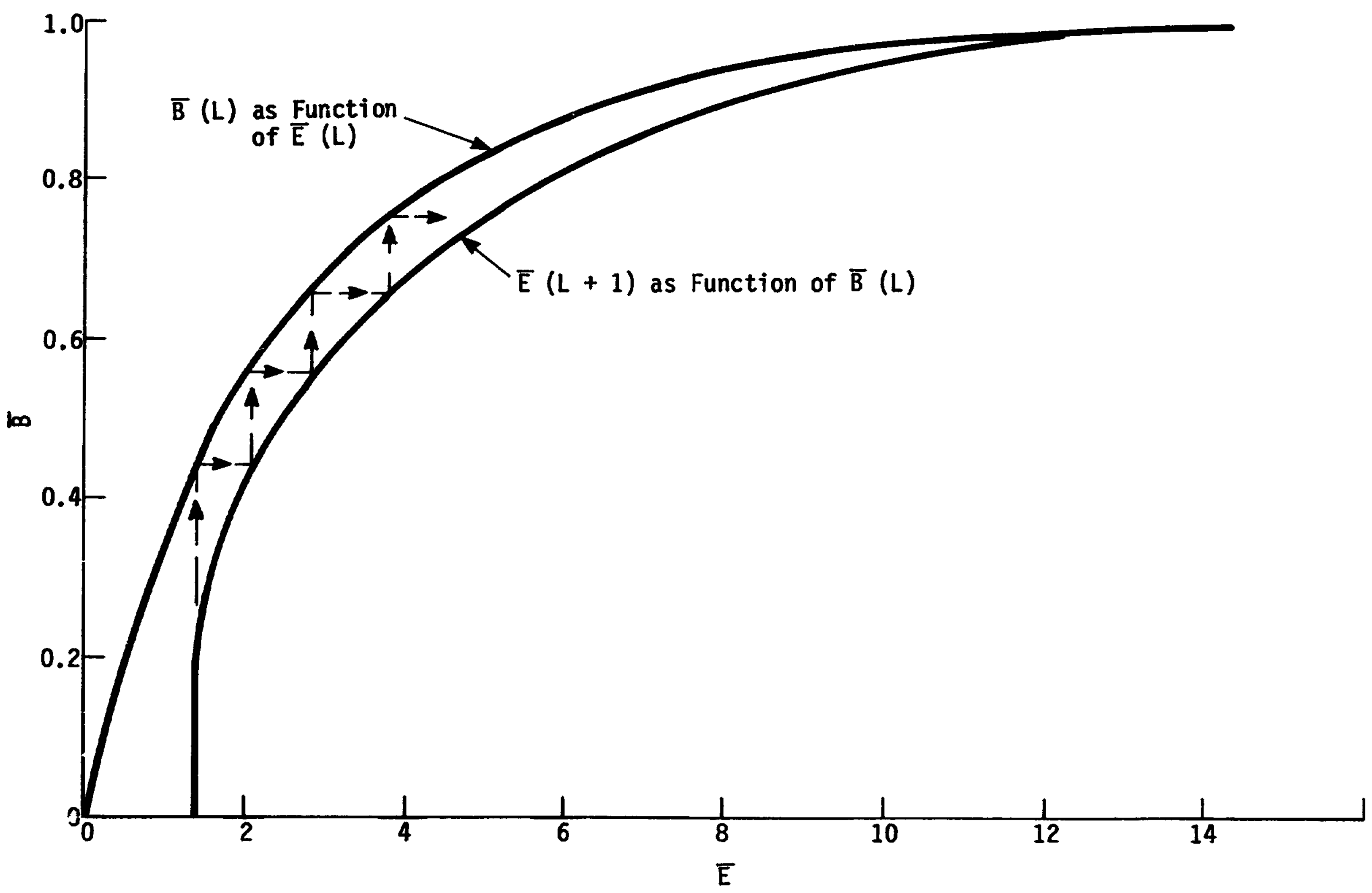


Fig. IV-1 Decoding Convergence

Thus, the information rate in bits/symbol, 1/V is

$$1/V = \frac{b^2}{2 \ln 2} = C, \qquad [IV-28]$$

the channel capacity.

From the foregoing it is deduced that it is possible to design a code that can be successfully decoded at any rate less than channel capacity. The number of decoding passes that are required will go up without limit as channel capacity is approached.

In applying this result to real channels and finite codes, keep in mind the assumptions that were made. The most restrictive assumption was that the B_i for any two bits are statistically independent. This is not true in a finite code. The estimates of bits sharing one or more channel symbols will be influenced together by the shared symbols and they will also influence each other. Therefore the code should be chosen to minimize the coupling between any small set of bits. Codes that satisfy this requirement will also have few words of minimum and near-minimum weight.

2. Channel Capacity

The capacity for a binary-input channel in the limit of low SNR is computed as follows.

The capacity C is equal to the input entropy (one bit per symbol) minus the uncertainty H introduced by the channel.

$$H = -\sum P\left[x_i\right]\left\{P\left[0|x_i\right]\log_2 P\left[0|x_i\right] + P\left[1|x_i\right]\log_2 P\left[1|x_i\right]\right\} \qquad [IV-29]$$

$$= -\sum P\left[x_i\right]\frac{\left(1+b_i\right)\ln\left(1+b_i\right)+\left(1-b_i\right)\ln\left(1-b_i\right)-2\ln 2}{2\ln 2} \qquad [IV-30]$$

Expanding ln $\left(1+b_i\right)$ in a power series gives

$$\ln\left(1+b_i\right) = b_i - b_i^2/2 + \ldots$$

$$\ln\left(1-b_i\right) = -b_i - b_i^2/2 - \ldots \qquad [IV-31]$$

Retaining the first two terms of Eq [IV-31] and substituting in Eq [IV-30] gives

$$H = 1 - \frac{1}{2\ \ln 2} \sum P \left[x_i\right] b_i^2 \qquad [IV-32]$$

$$C = 1 - H = \frac{1}{2\ \ln 2} \sum b_i^2\ P \left[x_i\right] \qquad [IV-33]$$

C. LOWER BOUND ON ERROR RATE

In decoding a given message bit, the number of symbol estimates that can be used is equal to the total number W of connections to the encoding shift register. Under the best circumstances, all the other bits that contributed to these symbols have been decoded correctly and have high estimates so that the decoder has high confidence in their correctness. Each symbol estimate will then be used at face value, and the probability of error for the given bit will be determined by the signal energy available in the W symbols. Since the energy actually expended per bit is only that in V symbols, performance will be equivalent to that of an uncoded channel operating at a signal-to-noise ration W/V times as high.

This bound is useful in predicting performance at high signal-to-noise ratio where the error rate is low and considerable simulation time is needed to get good statistics. Since the bound is a good approximation to the real behavior above some critical signal-to-noise ratio simulation is only needed to delineate the way in which the bound is approached.

D. SIMULATION RESULTS

The new algorithm was implemented in FORTRAN on a CDC 6400 computer. A random number generator was used to determine the symbol estimates with either three- or four-bit quantizing. The function F(x) defined in Eq [IV-9] was obtained from a look-up table with 64 entries.

To avoid end effects, we turned the convolutional code into a cyclic block code. Mathematically, this can be described by stating that bit index I is to be taken modulo Z where Z is the number of message bits in a block.

In terms of a shift register encoder, it means that the first K message bits are loaded into the register before starting to generate channel symbols, and when the end of the block is reached, the first K-1 bits follow the last bit into the register. Exactly ZV symbols are generated, every bit contributes to the same number of symbols and has an equal chance of correct decoding, and there is no tail of check symbols at the end of the block.

The computer handles an entire block at a time, and it does not care how large the convolutional constraint length K is. Therefore, we used values of K that nearly fill the block, and performance is limited by block length rather than constraint length. The particular codes were chosen with the help of the tables of Robinson and Bernstein* with additional space between the connections to increase K.

Figure IV-2 shows results obtained with a V = 3 code. The shift register connections defining the code are 1; 1,205,367; 1,6,29,216,217,567,591,938,958. Received symbols were quantized to 16 levels. The block length is 2000 and the iteration limit 20. At E_b/N_o = 1.3 dB, 739 errors remained in five blocks (10,000 bits). At 1.5 dB only a single error remained in five blocks, and at 1.7 dB there were three errors in seven blocks. To compare performance with sequential decoding, the computational limit was calculated from the transition probabilities actually used for the 1.5 dB simulation, and it was found to be 0.297 bit/symbol. At V = 3, we are transmitting 0.333 bit/symbol, or 12% more than would be possible at the computational limit for sequential decoding. Since the new algorithm is self-starting, it does not require a periodic interruption of the message with a restart sequence as does sequential decoding.

Channel capacity, the ultimate theoretical limit, was also calculated from the transition probabilities at 1.5 dB and found to be 0.463 bit/symbol. At 0.333 bit/symbol we are operating at 72% of channel capacity.

A word of caution is required in interpreting the significance of the simulation. Each run contained only five or seven blocks, and a single bad block can have many errors. The curve that has

*John P. Robinson and Arthur J. Bernstein: "A Class of Binary Recurrent Codes with Limited Error Propagation." *IEEE Transactions on Information Theory*. Vol IT-13, No. 1, January 1967.

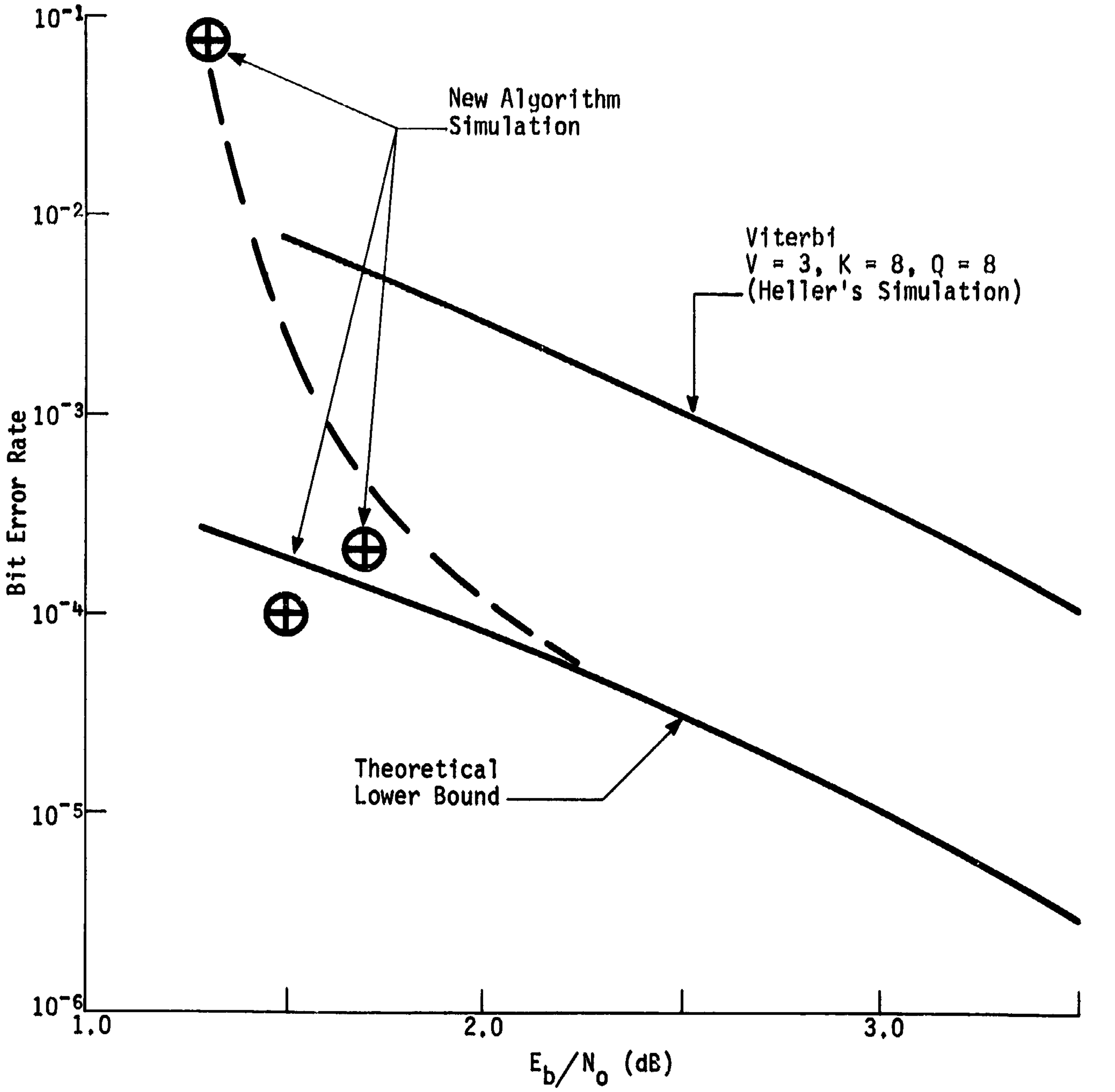


Fig. IV-2 Performance of New Algorithm with V = 3, Q = 16, and Block Length = 2000.

been drawn to suggest how the performance trends into the theoretical lower bound (explained in the previous section) is therefore only approximate.

Figure IV-3 shows how decoding proceeded for the V = 3 code at 1.3 dB and 1.5 dB. For 1.3 dB it is seen that one block decoded quickly to a low error count, although it finished with three errors. Two other blocks were still improving when processing was stopped after 20 iterations, and the two worst blocks never got a decent start and are apparently getting slightly worse. At 1.5 dB from 10 to 18 iterations were needed for a perfect result, and the block that retained one error stabilized at this level on the 17th iteration. At 1.7 dB the seven runs stabilized at their final error counts in from seven to 13 iterations. This rapid convergence increases confidence that there is a low probability of really bad blocks at this signal-to-noise ratio.

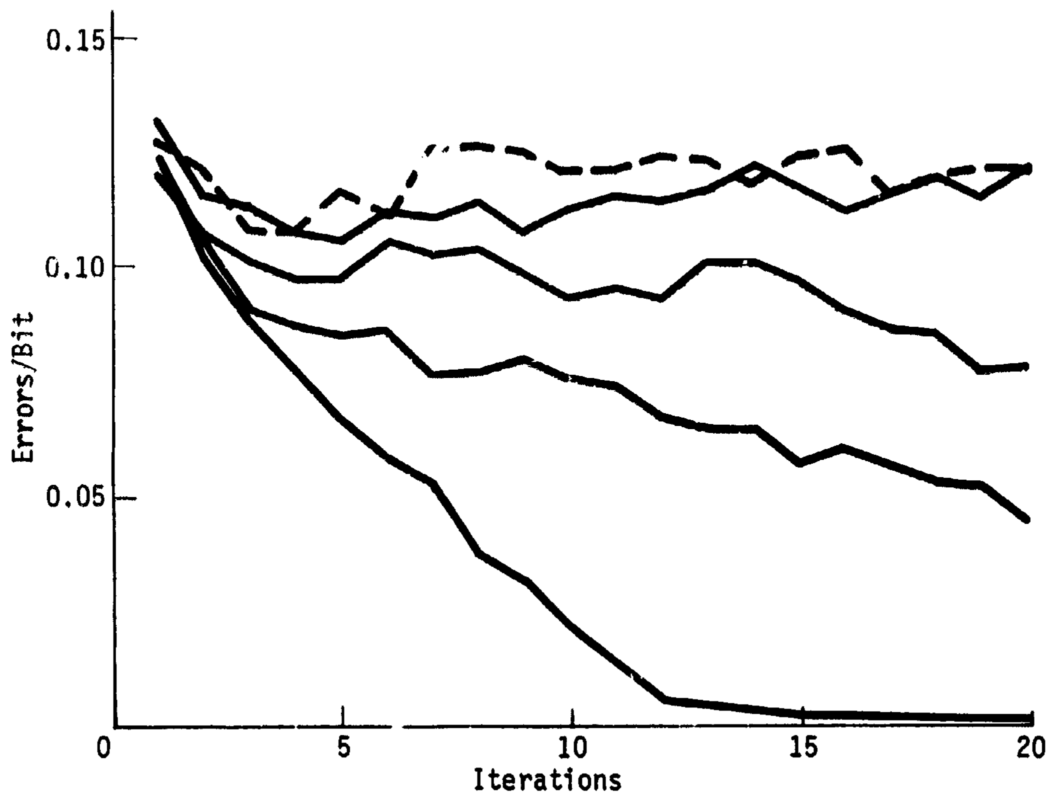
Figure IV-4 shows the results of simulation of the new algorithm with V = 2, eight levels of quantizing, and a block length of 500. Each run represents 10 blocks. The run at 3.3 dB had no errors in 5000 bits. Performance curves for Viterbi decoding and for hard-decision Fano decoding are shown for comparison. Simulation has demonstrated the following properties of the new algorithm:

- For V = 3 it will perform satisfactorily at rates substantially higher than the computational limit for sequential decoding;
- It is self-starting;
- Large constraint lengths are practical.

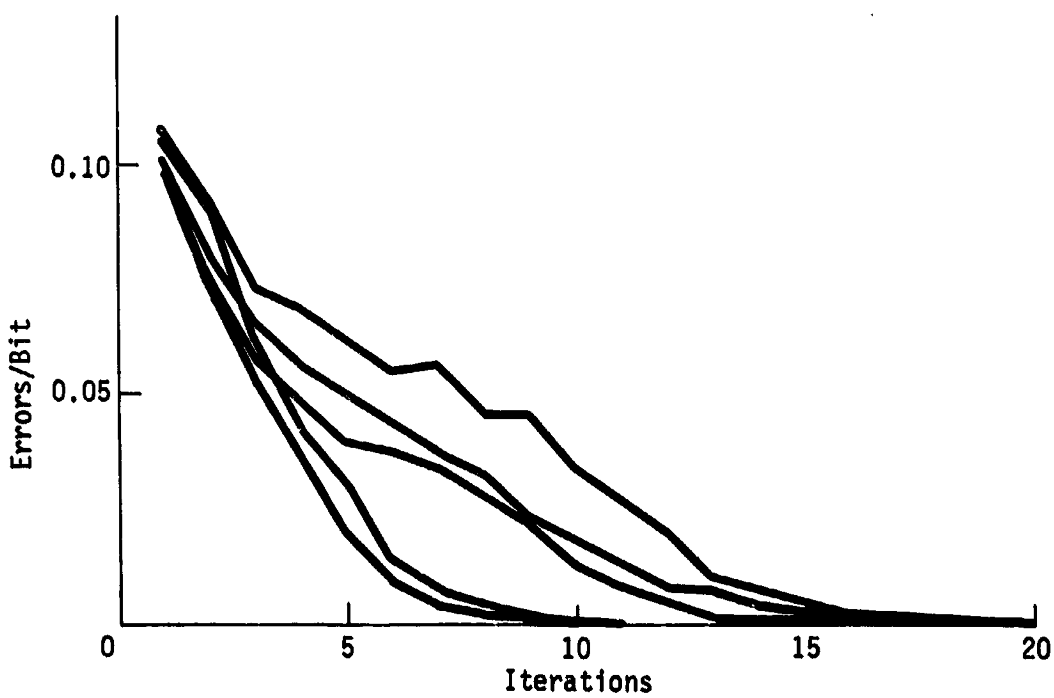
Further work should be devoted to these areas:

- Simulation at V = 4 and greater;
- Improved simulation statistics;
- A better theoretical understanding at finite V and K;
- Delineation of the exact effect of K on performance.

Chapter V describes a breadboard decoder for a small code and a single iteration that operates at message rates of more than 10 megabits/sec.



(a) E_b/N_o = 1.3 dB



(b) E_b/N_o = 1.5 dB

Fig. IV-3 Decoding Progress of the New Algorithm with V = 3, K = 958, and Blocks of 2000 Bits.

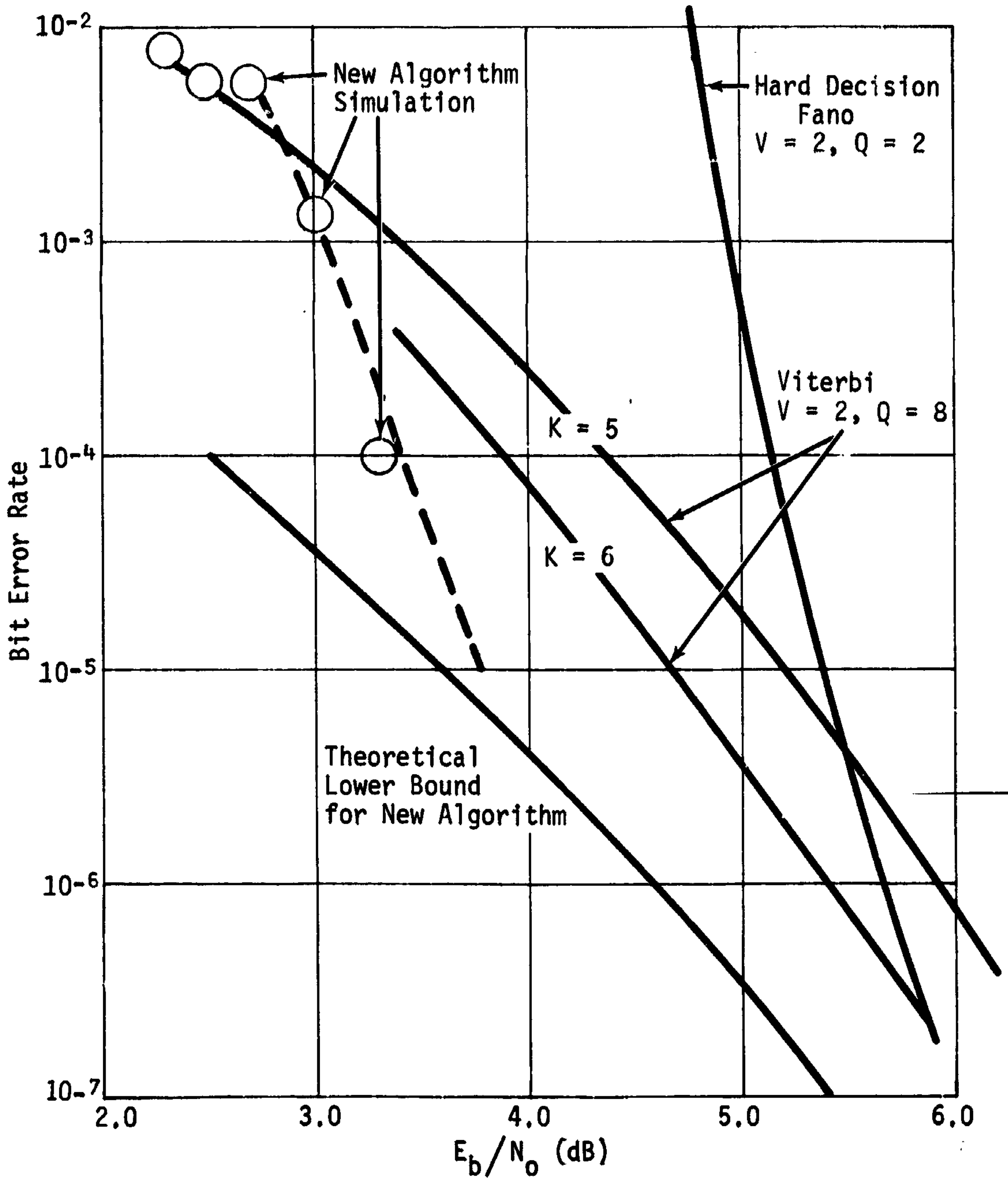


Fig. IV-4 Performance of New Algorithm with V = 2, Q = 8, and Block Length = 500.

E. SIMULATION OF THE NEW DECODING ALGORITHM

The simulation approach and notation used in the simulation program are delineated in this section and Fig. IV-5.

To do the simulation of the decoding process a set of values of Gaussian distribution are computed as decoder inputs. These values are then quantizes to eight or 16 equally spaced levels. For each quantization level the logarithmic a posteriori probability ratio $\ln \frac{P(0|X_k)}{P(1|X_k)}$, where X_k is the k^{th} quantization value, is computed. For each decoder input this ratio is computed and used for decoding. In the program these values are called;

$$WQ(I,J),\ 1 \leq J \leq V$$
$$1 \leq I \leq \text{Block Length}$$

For each message bit encoded there will be V code symbols transmitted (convolutional encoding). The number of these sets of V elements (i.e., the message length) to be operated on in the simulation is given by the block length, this is denoted by BL in the program. A particular set of code symbols is indexed by the subscript I. Code symbols for the I^{th} message bit are indexed by the subscript J. The first of these (J = 1), is the message bit being encoded (systematic bit). The information rate transmitted is given by the ratio, R = 1/V where V is the number of code symbols for each message bit.

During each iteraiton of the decoding scheme a modified value for the logarithmic likelihood ratio given in the preceding paragraph (WQ(I, J)) is computed. After all iterations are completed a decision based on the message estimate at that time is made. To compute the modified value while decoding a particular bit, we perform the inverse mod-2 operation (i.e., $A = B \oplus C \rightarrow B = A \oplus C$) which this bit entered into in the message encoding process. This is done with respect to each of the code symbols influenced by the particular message bit being decoded. Then the likelihood opinion of the message bit is revised by adding (mod-2) to it the results of these inverse mod-2 operations.

To perform the process of the preceding paragraph, the simulation program enters the computed WQ(I, 1) values as the first message estimate, ME(I, 1). The length of the encoder register i.e., the constraint length, is given by K in the program.

Hence, in the encoding-decoding process any message bit can be affected by other bits a distance of K message bit positions on either side, and any bit can effect KV transmitted digits.

In the decoding process the likelihood estimate of the message bit being decoded WQ(I, J) is revised. This is done by taking the sum of the ln TANH[MF(I, J)] (this function corresponds to the mod-2 addition) of all other code symbols into which the message bit entered. In the simulation this sum is called MEP and its terms are given by MEF(I, FIRST), the sign of MEF is called MEFS. When this sum is attained for each connection vector these sums (subtotals) are added (mod-2) with WQ(I, J). In the simulation this is actually done by first summing ln TANH(X) of all those WQ(I, J) which are of positive signs and then subtracting from this MEP the MEF(I, FIRST) corresponding to the bit being decoded. The inverse function is then obtained in the look-up table. This process is gone through for each bit in the block for each connection vector (mod-2 adder) of the encoder. The number of times this process is repeated is limited by IL (iteration limit). The connection vectors are given by C(I, J) in the program. The number of blocks to process in a run is set by the input parameter PASS.

Upon reaching the bit string end for an iteration there will be some inverse mod-2 operations not yet performed for the last transmitted bit. To handle these effects the simulation uses two options established by the input parameter S3. If S3 is set to 1 the end effects are ignored. If S3 is set to 2 a "wraparound" approach is used bringing around bits from the beginning of the bit string.

In the simulation a message of all zeros is assumed; hence any negative values remaining in the ME at the end of the run are counted as errors.

ALGORITHM VARIBLES:

```
WQ(I, J)   RECEIVED QUANTIZED BIT REGISTER
      I  = INPUT BIT POSITION
      J  = RATE DENOMINATOR
ME(I,FIRST) MESSAGE ESTIMATE (DOUBLE BUFFERED)
      I  = COUNTER OF DECODING POSITION
      J  = RATE DENOMINATOR (WHICH CODE VECTOR)
      L  = ITERATION COUNTER
      N  = CONNECTION SUBITERATION WITHIN CONNECTION VECTOR-
           NOTE:  NOT A BIT POSITION INDICATOR
FIRST, SEC = DOUBLE BUFFER POINTER (REPLACES L FOR ACTUAL)
    MEP  = MESSAGE ESTIMATE PRODUCT - USED ONLY FOR METHOD OF
           CALCULATION OF INDIVIDUAL = TOTAL MINUS OTHERS.
MEF(I,FIRST) MESSAGE ESTIMATE FUNCTION
```

INPUT PARAMETERS:

```
       S1  = FOR INDEXING RATE NUMERATOR, S1=1 FOR REGULAR IN-
             DEXING, S1=2 FOR ALTERNATING RATE NUMERATOR
       S2  = CHOICE OF BIT INPUT S2=1 FOR INFINITE BIT STRING,
             S2=2 FOR STATIC FILL
       S3  = WRAP AROUND ENCODER INPUT S3=1 IGNORE END EFFECTS,
             S3=2 TO WRAP AROUND END.
       S4  = CHOICE OF LOOKUP TABLES 1 FOR 32, 2 FOR 64.
       S5  = IMPROVEMENT BY DEGRADING BY SUBTRACTIVE FACTOR-ON
             ALL MES BY USING A=(MES-WQA(1))/SUMMATION T(V)
       S6  = COLLECTION AND PRINTOUT OF ME STATISTICS IN FORM
             OF LN N(I+)/N(I-)
        P  = PRINT OPTION INDEX, ALL LEVELS INCLUDE THOSE ABOVE
             ITS OWN LEVEL.  (SEE LISTING)
       BL  = BLOCK LENGTH
        K  = CONSTRAINT LENGTH
        V  = RATE DENOMINATOR
   C(V,M)  = ACTUAL CONNECTION AND BIT POSITION
     T(V)  = NUMBER OF MOD-TWO ADDERS PER CONNECTION VECTOR
       IL  = LIMIT TO NUMBER OF ITERATIONS
       NI  = NUMBER OF BITS TO INSERT WITH EACH ITERATION
     PASS  = NO OF BLOCKS TO PROCESS
 QUANT(I)  = QUANTIZATION TABLE BREAKPOINTS
    START  = WHICH ITERATION NUMBER TO START TO CALCULATE IM-
             PROVEMENT - SEE S5 OPTION CONTROL
```

Fig. IV-5 New Decoding Algorithm Simulation

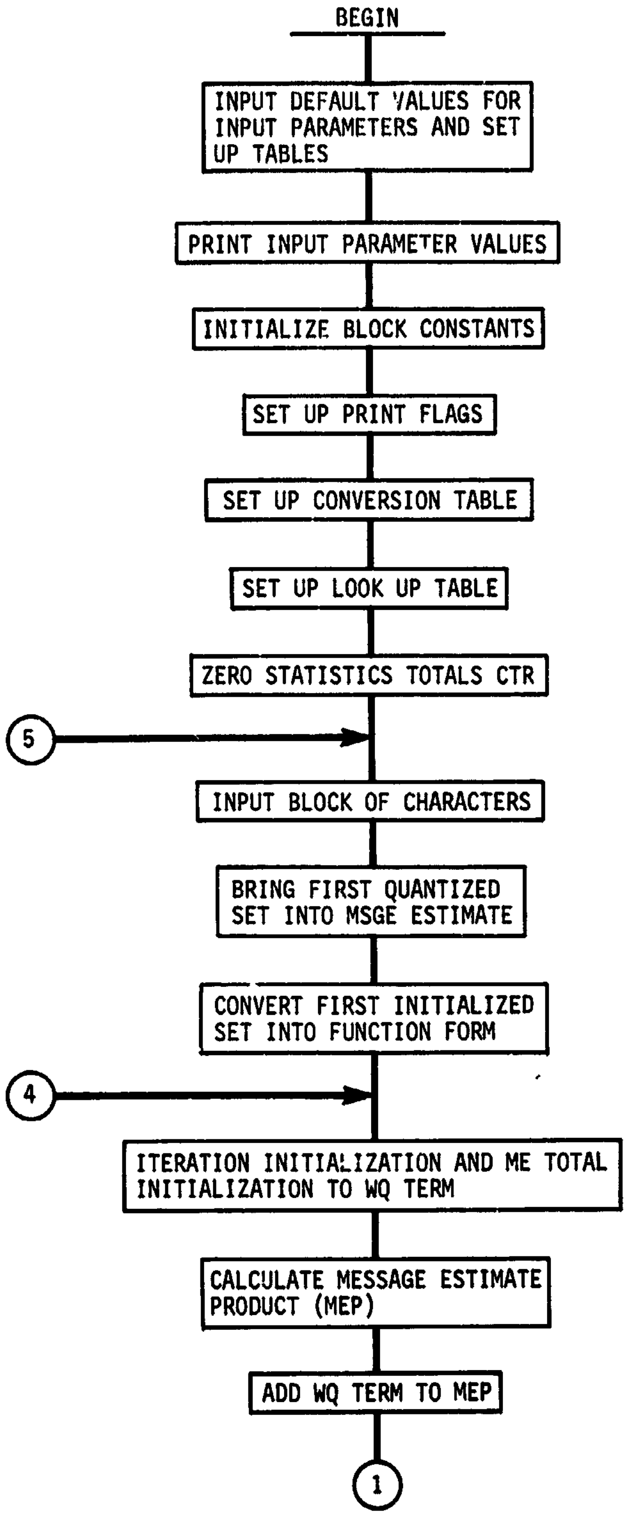

BEGIN
INPUT DEFAULT VALUES FOR INPUT PARAMETERS AND SET UP TABLES
PRINT INPUT PARAMETER VALUES
INITIALIZE BLOCK CONSTANTS
SET UP PRINT FLAGS
SET UP CONVERSION TABLE
SET UP LOOK UP TABLE
ZERO STATISTICS TOTALS CTR
5
INPUT BLOCK OF CHARACTERS
BRING FIRST QUANTIZED SET INTO MSGE ESTIMATE
CONVERT FIRST INITIALIZED SET INTO FUNCTION FORM
4
ITERATION INITIALIZATION AND ME TOTAL INITIALIZATION TO WQ TERM
CALCULATE MESSAGE ESTIMATE PRODUCT (MEP)
ADD WQ TERM TO MEP
1

Fig. IV-5 (cont)

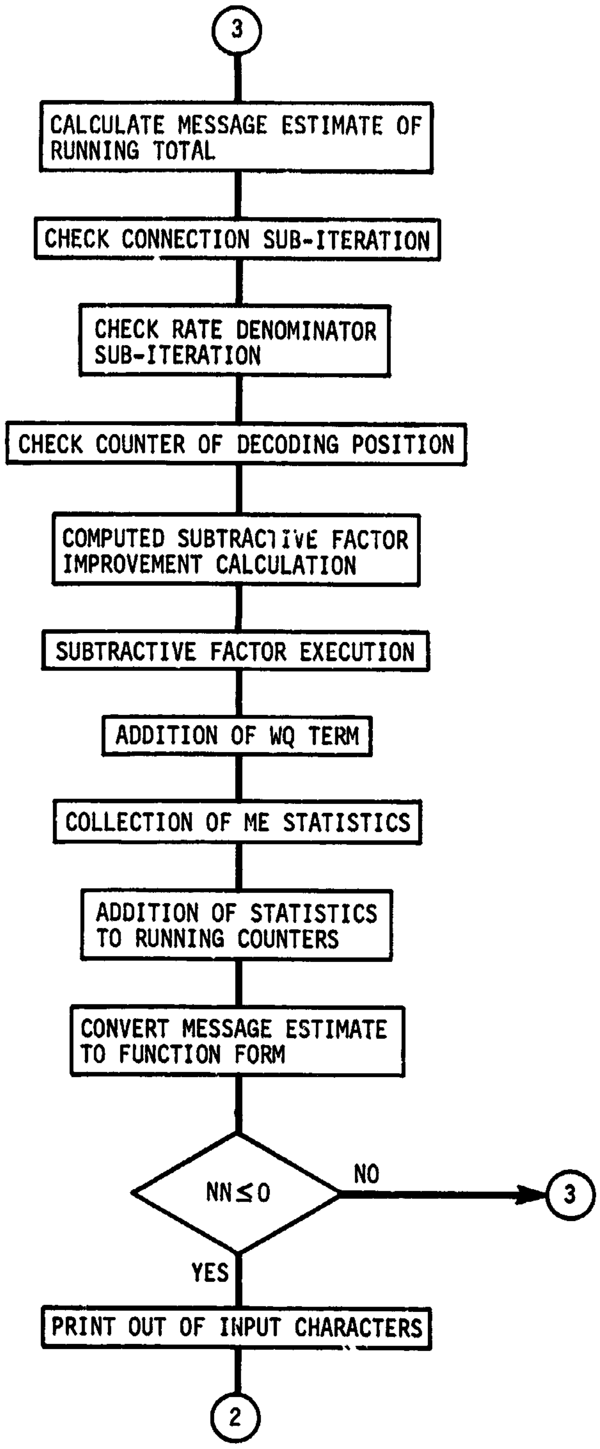
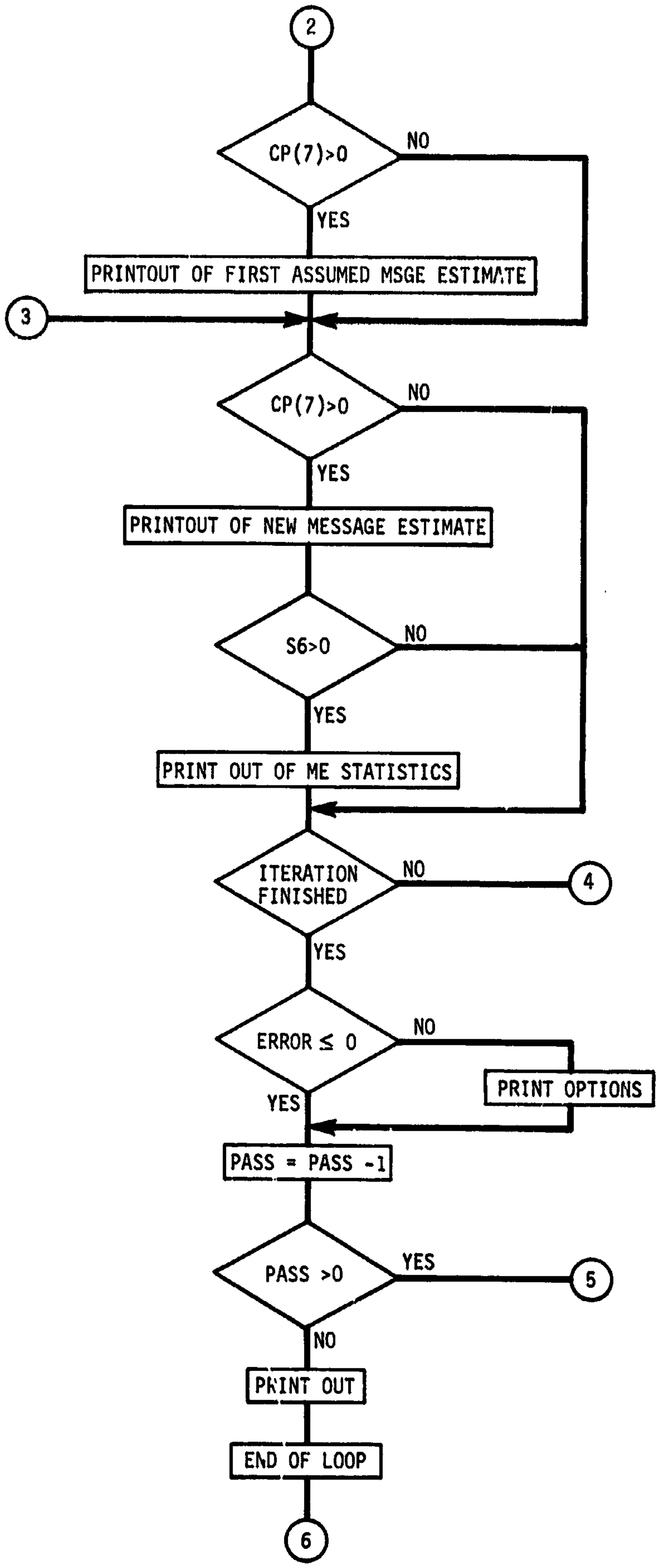

3
CALCULATE MESSAGE ESTIMATE OF RUNNING TOTAL
CHECK CONNECTION SUB-ITERATION
CHECK RATE DENOMINATOR SUB-ITERATION
CHECK COUNTER OF DECODING POSITION
COMPUTED SUBTRACTIVE FACTOR IMPROVEMENT CALCULATION
SUBTRACTIVE FACTOR EXECUTION
ADDITION OF WQ TERM
COLLECTION OF ME STATISTICS
ADDITION OF STATISTICS TO RUNNING COUNTERS
CONVERT MESSAGE ESTIMATE TO FUNCTION FORM
NN≤0
NO
3
YES
PRINT OUT OF INPUT CHARACTERS
2

Fig. IV-5 (cont)


2
CP(7)>0
NO
YES
PRINTOUT OF FIRST ASSUMED MSGE ESTIMATE
3
CP(7)>0
NO
YES
PRINTOUT OF NEW MESSAGE ESTIMATE
S6>0
NO
YES
PRINT OUT OF ME STATISTICS
ITERATION FINISHED
NO
4
YES
ERROR ≤ 0
NO
PRINT OPTIONS
YES
PASS = PASS -1
PASS >0
YES
5
NO
PRINT OUT
END OF LOOP
6

(Fig. IV-5 (cont)

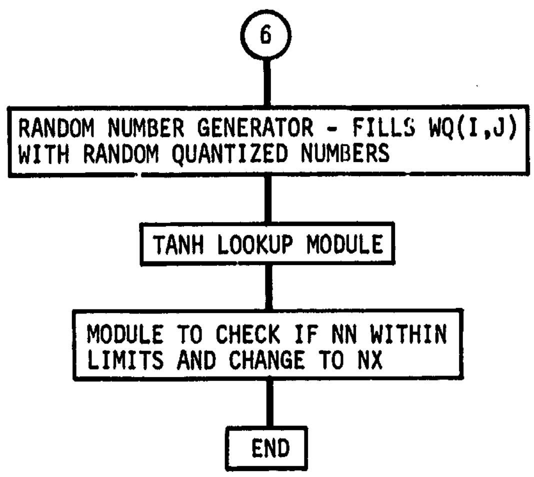


Fig. IV-5 (concl)

V. HIGH-SPEED DECODER

A. DESIGN OBJECTIVES

The basic objective of this task was to produce a high-speed demonstrator of the capabilities of the new decoding algorithm.

The high-speed requirement is imposed to demonstrate the capability at any speed. That is, if the decoder will perform at high speed it can be made to perform at lower speed with little additional effort and ordinarily with less hardware expense.

Originally only certain sections of this decoder were to be built. However, it was soon discovered that designing the auxiliary equipment necessary to demonstrate only the critical sections of the decoder would be a greater task than designing an entire encoder/decoder system on a small scale. Therefore, a complete self-contained demonstrator was produced except for power supply and an oscilloscope to display the operation.

Primary objectives of maximum speed at minimum costs have been attained. The demonstrator contains its own message source and noise source and clock. But the capability of supplying external data and noise and the clock are provided through front panel connections.

Rather than attempt to set a bit rate requirement on the demonstrator, and attempt to achieve it with possible failure, the configuration of the decoder was allowed to dictate the maximum bit rate capability. This allows the family of logic and the design technique to dictate the maximum bit rate operation.

B. GROUND RULES

The ground rules for this demonstration decoder were established at the outset of the design stage and were carried out throughout the design:

1) Produce a simplified implementation of the new decoding algorithm that will demonstrate a larger decorder's capabilities at minimum hardware expense;
2) Demonstrate maximum bit rate capability for a large decoder system with a small system;
3) Maximum use of MSI components wherever possible and use of LSI if the particular elements are available;
4) Worst-case design rules are to be used throughout;
5) 74H Series TTL is to be used in all cases where individual logic functions, latching, and gating, are necessary;
6) Lower speed logic will be used wherever possible in the man/machine interface;
7) Full synchronous sequential design is to be used throughout the decoder;
8) The decoder is not designed to gather performance data -- this task is left to the general purpose computers.

C. DESIGN

The following is a functional description of each module as it appears in Fig V-1 thru V-25.

1. System Block Diagram

Figure V-1 is a system block diagram showing the hardware layout of the demonstration decoder. This figure can be used for referencing the positions of the other 24 figures that follow, and is helpful in going through the description of the function of each of these modules.

2. M Register and Encoder

The first module encountered in the demonstrator is called the M register (for the message register) and encoder shown in Fig. V-2. This is a high-speed convolutional encoder, very similar to that found at the transmitting end of an operational error correction system. This unit consists of a 16-bit flip-flop register. The most significant six bits, or the left-most in this drawing, are the encoder register. The block shown directly below this register is the actual mod-2 adder that produces the transmitted symbol.

Also included in this drawing is the hardware that creates the quantized outputs from the encoder. These quantizers receive their noise information from a set of four noise registers shown in the next four drawings that simulate the space channel. Thus the effective outputs of this module are WQ terms.

The operation of this modul is as follows: There are two entry ports for data into this module and two entry ports for noise. The data may be entered through two pushbutton switches for 1's and 0's, and these pushbuttons must be depressed to produce the 16 message bits that are encoded. The other input port is termed "external M," as shown at the top of Fig. V-2, and is the port by which external data may be applied to this system. The normal mode of operation for demonstration will be to use the pushbutton switches to enter 16 bits of data. When loading these 16 bits of data, the operator must be careful because the first bit that is loaded will not be the first systematic bit that is encoded. The first bit loaded will actually be the most significant bit in the register. The sixth bit that is loaded into the register will be the first bit that will be presented to the encoder for encoding as the systematic bit. The first bit that is loaded will have an effect on this bit in that it is under the encoder at the time of encoding. This may seem confusing at first, but a little practice in operating the system will eliminate the confusion that occurs.

The actual procedure for loading is to depress either a 1 or a 0 button indicating the bit that is to be loaded. When this is done, the value of the bit is impressed through the direct set or clear inputs of the flip-flop and a manual clock pulse is generated, which shifts this bit over to the next flip-flop, thus shifting the entire register one bit over. This bit is loaded into the most significant bit position and then shifted, which makes it resident in the least significant bit position. By the time 16 bits are loaded, this bit will again be in the most significant bit position due to the shifting through the register.

The connections for the code are hardwired to the module directly below the register, Part No. N8262A. This is a parity generator or mod-2 adder, which creates the encoded word for the nonsystematic (or check) vector. Noise is applied to these two bits that are created by the encoder and the outputs are then converted to WQ sign 1, WQ magnitude 1, WQ sign 2, and WQ magnitude 2 and their complements, which are the outputs of this module. This module is also furnished with a system clock so that it is synchronous with the entire system.

There is a toggle switch on the front panel that allows selection of internal or external data. This switch also governs internal and external noise through the gates across the bottom of the figure.

3. Noise Input Registers

Figures V-3 thru V-6 show the four noise input registers. These are very similar to the M register and are loaded in an identical fashion to the M register. Noise is not loaded into these registers from an external source during the external mode of operation because it is not necessary to store the noise that is coming in from an external source as it is with the message. Pushbuttons on the front panel are identical to those which load the M register. There are four pairs of these for loading noise and the same rules that the sixth bit loaded is the first bit encoded hold for these four units. Once again, although this seems rather difficult at first, a small amount of practice using these loading techniques will make the procedure very simple. The reason this procedure was chosen is to avoid the necessity for five sets of 16 switches to load these registers.

4. WQ Registers

The Word Quantized (WQ) registers shown in Fig. V-7 are actually the first module of the decoder, the previous boards being the encoder portion of the system. This WQ register would normally receive its input WQ's and their complements from an analog-to-digital (A/D) converter that would have quantized the received signal. In this case the A/D converter is simulated by the previous modules where the noise is added mod-2 to the message. This is done because the demonstrator never creates an analog signal as such. These WQ's are loaded into the first flip-flops on the right, simultaneously, and are picked up by the rest of the system through modules that will be explained later. Notice that WQS1 and WQM1 are held for five bits or five clock times and that the WQ2's are not. This is because these bits are not used until later when they are quantized by the circuit on the left hand section of the drawing, the logic and flip-flops shown there, create the transformed word quantized or as shown TWQO through TWQ3 and TWQS. These bits are then used in the arithmetic units as decribed later.

Once again, this module looks exactly as it would in a functioning decoding system, although it is small compared to what it would be in a more highly refined system.

5. MEF Registers

The modules shown in Fig. V-8 thru V-11 are the MEF registers signifying the Message Estimate Function registers. These registers are initially preset to all 1's at the outset of a decoding cycle to neutralize the data left in them. There is a preset pushbutton on the front panel. These registers furnish data about the previous bits of the system. It is difficult to explain this register at this time, because nothing has occurred that loads this register except the preset. On subsequent iterations these four registers would have been reloaded from the signal RMEF or Resultant Message Estimate Function from a previous iteration cycle and would now be impressing data about the previous iteration on the present iteration of data.

6. Subtotal Adders and MEPF

Figure V-12 shows the subtotal registers and adders. This module receives its data from WQ and MEF creating and storing a subtotal in an add cycle. In other words WQ2 is added to MEFO and MEF2 is added to MEF5 and their sums are stored. The signs from WQ2 and MEFO are mod-2 added and stored; also the signs MEF2 and MEF5 are mod-2 added and stored, creating a subtotal that is presented to the module of Fig. V-13 where the Message Estimate Product Function (MEPF) is generated as a sum of subtotal 1 and subtotal 2. These are then stored in the flip-flops. Subtotal 1 sign and subtotal 2 sign are mod-2 added and stored. Each of these steps now have been proceeding at one bit time per operation.

7. MESF

The data from the MEPF now are distributed to three subtractor units that create the Message Estimate Subtotal Function (MESF). These three subtractors are shown in Fig. V-14, V-15, and V-16. Their function is to take the MEPF and subtract from it the MEF of the bit of interest, also performing mod-2 subtraction on the sign bits and storing these.

8. MESO, MES2, and MES5

Figures V-17, V-18, and V-19 show the Message Estimate Subtotal modules that are entitled MESO, MES2, and MES5.

2

These three modules receive data from the MESF modules and transform these data by the function defined in equation [IV-9] of Chapter IV. The outputs of this function transform are then latched in flip-flops as shown and held for use by the next set of modules.

The function performed by this module is shown in Table V-1 and should completely explain the operation of this module.

Table V-1 Module Function

	Input x	Output f(x)
0	0 0 0 0	1 0 1 0
1	0 0 0 1	0 1 1 0
2	0 0 1 0	0 1 0 0
3	0 0 1 1	0 0 1 1
4	0 1 0 0	0 0 1 0
5	0 1 0 1	0 0 0 1
6	0 1 1 0	0 0 0 1
7	0 1 1 1	0 0 0 0
8	1 0 0 0	0 0 0 0
9	1 0 0 1	0 0 0 0
10	1 0 1 0	0 0 0 0
11	1 0 1 1	0 0 0 0
12	1 1 0 0	0 0 0 0
13	1 1 0 1	0 0 0 0
14	1 1 1 0	0 0 0 0
15	1 1 1 1	0 0 0 0

9. Arithmetic Units

The arithmetic units are shown in Fig. V-20, V-21, and V-22. They are simply explained as full adder-subtractor modules, receiving their inputs in the case of Fig. V-20, from MESO and TWQ. Inputs to Fig. V-21 are from MES2 and MERT1, and inputs to Fig. V-22 are MES5 and MERT4.

Depending on the sign and magnitude of the inputs to these modules, they produce either an addition or subtraction, combining the two input numbers and produce the resultant output.

Inputs to these modules are presented in 2's complement form, and outputs are generated in 2's complement form.

10. MERT

The Message Estimate Running Total (MERT) module of Fig. V-23 is the holding register for the arithmetic manipulations. It is basically a sequential accumulator that spaces the message estimate running totals for use by the adders described in the previous section.

The sums must be held for specific bit periods to synchronize them with the other operations of the decoder.

11. RMEF

The Resultant Message Estimate Function (RMEF) module shown in Fig. V-24 is the final unit of the decoder and consists of the output function generator, the output unit, and some housekeeping hardware that reorients the data for use by subsequent iteration cycles.

The inputs to this module are MERT5, coming from the last arithmetic unit of the decoder. This module first takes the 2's complement resultant of the MERT5 and puts this in sign magnitude form. This sign magnitude number is then latched and at the next bit time is presented to the output function generator which produces the RMEF. The resultant Message Estimate (ME) is also output by this module.

The data RMEF are presented back to the MEF registers when the switch on the front panel, called ITERATE, is depressed. These data are presented for one period of data or 16 bits, then data into the MEF are stopped, and the decoder goes back into the display mode. The ME output is continuously presented to the front panel for display on the oscilloscope.

12. Timing and Control Logic

Figure V-25 shows the timing and control logic for the entire demonstrator. It consists of an external clock-entry port with associated electronics to square up this input signal and format it as a clock for the system.

Also included in this module is the internal crystal oscillator module with its squaring electronics and the countdown register to produce the system clock.

The ITERATE switch and the RUN/STOP switch that control the operation of the decoder along with the 16-bit synchronizing counter are included. The primary output of this module is CLOCK', which is distributed to each of the other modules of the system, and the SYNC output for synchronizing the display on the oscilloscope.

This concludes the detailed description of each of the modules included in the demonstration decoder. Photographs of the finished demonstrator are shown in Figs. V-26, V-27 and V-28.

D. OPERATIONG PROCEDURE

A step-by-step operating procedure that must be followed to properly operate the demonstration decoder is presented in this section. Many of the steps given here must be followed in order, and the specifications given must be closely adhered to because of the critical nature of some of the components used in this system. Because this is a demonstration breadboard, many of the safeguards for overvoltage and overcurrent have not been included.

The procedure is as follows:

1) Secure a power supply of nominal 5 v output, capacity 10 amps. This voltage must be well regulated and should not exceed 5½ v nor be less than 4½ v. This is a critical parameter, and the demonstrator should be operated as nearly at 5 v as possible. The power supply voltage should be checked before connection is made to the front panel jacks for power. This will guarantee that the logic of the system will not be damaged, as there is no overvoltage protection and there is no power switch included in the demonstrator itself;

2) An oscilloscope of the Tectronix 585 type or Hewlett-Packard 180 should be used with this demonstrator. These oscilloscopes provide sufficient bandwidth to demonstrate the capabilities of this unit. The trigger

input to the oscilloscope should be connected to the sync output on the front panel of the demonstrator. The leading edge of the synchronization pulse should be set to the left side of the graticule of the oscilloscope screen. The total cycle width of the synchronization pulse should be set to 8 cm. This allows one to determine which bit of information is being looked at in that each of the 16 bits include 1/2 cm of graticule;

3) After the power supply has been connected and the synchronizing output has been connected to the oscilloscope the RUN/STOP switch should be set to the STOP position, and the PRESET button should be depressed once. In the STOP position, the front panel data and noise pushbuttons are enabled. Data can now be inserted into the message register and noise into the noise registers;

4) Data and noise are entered through the 10 pushbuttons on the front panel. Each time a button is depressed a 1 or 0 is loaded into the next most significant bit of the register, depending on which pushbutton is depressed. Care must be exercised in loading data and noise because the sixth bit loaded into the registers becomes the first systematic bit of the encoder cycle. After loading the message and noise the RUN/STOP switch should be put to the RUN position. Data are now being transmitted and received;

5) Operation is ovserved by using a dual trace plug-in in the oscilloscope set to the alternate mode of operation. The inputs should be M and ME from the front panel of the decoder. These will show the original message (M) and what the decoder has estimated is the message (ME);

6) The ITERATE pushbutton may now be depressed and the decoder will make one iteration and give a better estimate of what the original message was. Each time the ITERATE pushbutton is depressed the decoder makes one iteration and returns to the display mode to continuously display the result;

7) The message received by the decoder may be observed as the 4 WQ outputs and the encoded message may be observed as M for the systematic bit and W2 for the nonsystematic bit;

8) External messages and noise may be presented to this demonstrator through the external connections. These signals should be 0 to 5 v TTL compatible NRZ signals synchronous with a sinusoidal clock signal supplied through the external clock connector. The outputs will be the same as for the internal data.

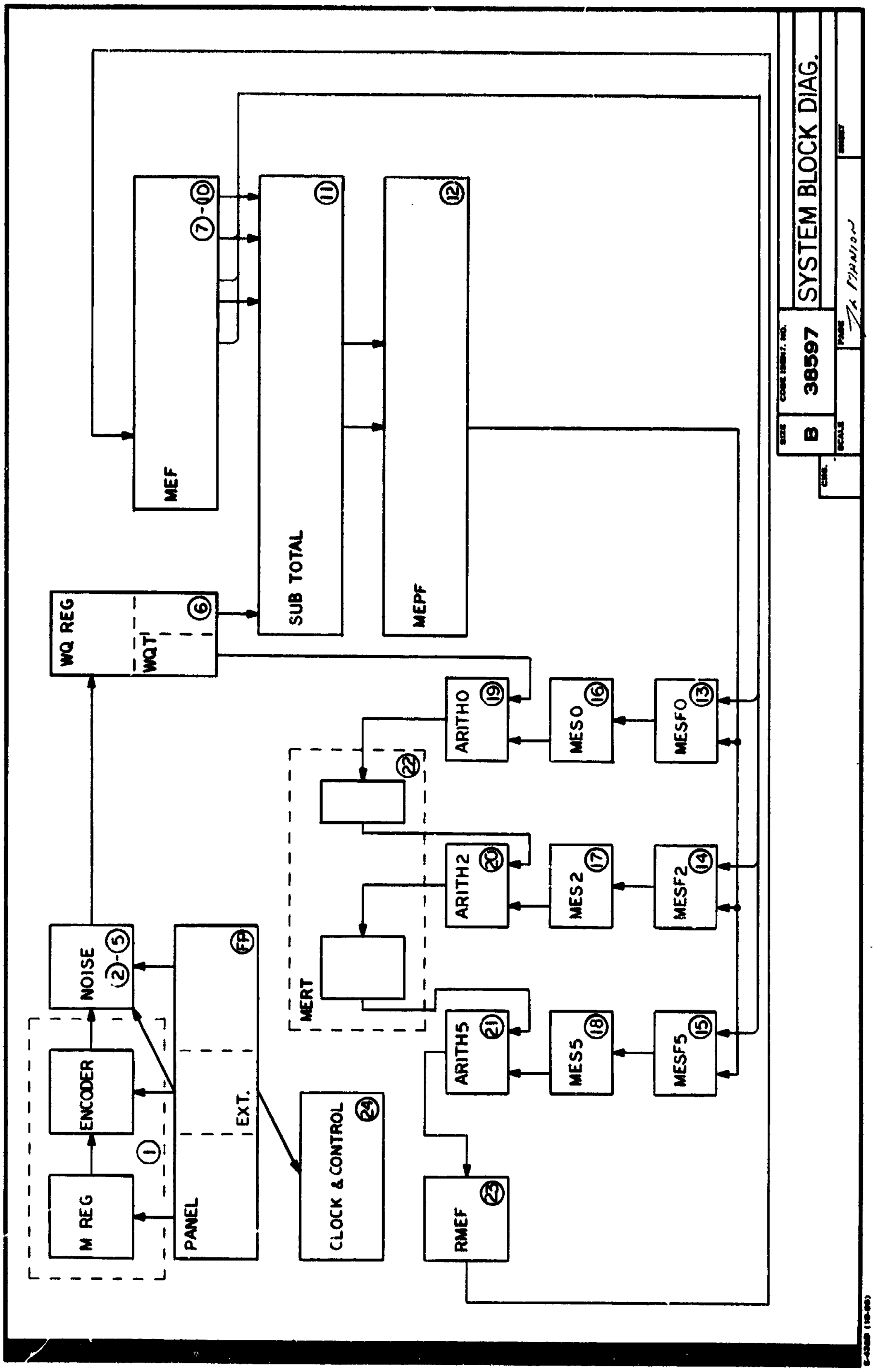

MEF
7-10
11
SUB TOTAL
12
MEPF
WQ REG
WQT
6
NOISE
2-5
ENCODER
M REG
1
PANEL
EXT.
FP
CLOCK & CONTROL
24
MERT
22
ARITHO
19
ARITH2
20
ARITH5
21
MESO
16
MES2
17
MES5
18
MESFO
13
MESF2
14
MESF5
15
RMEF
23
SYSTEM BLOCK DIAG.
B
38597

Fig. V-1

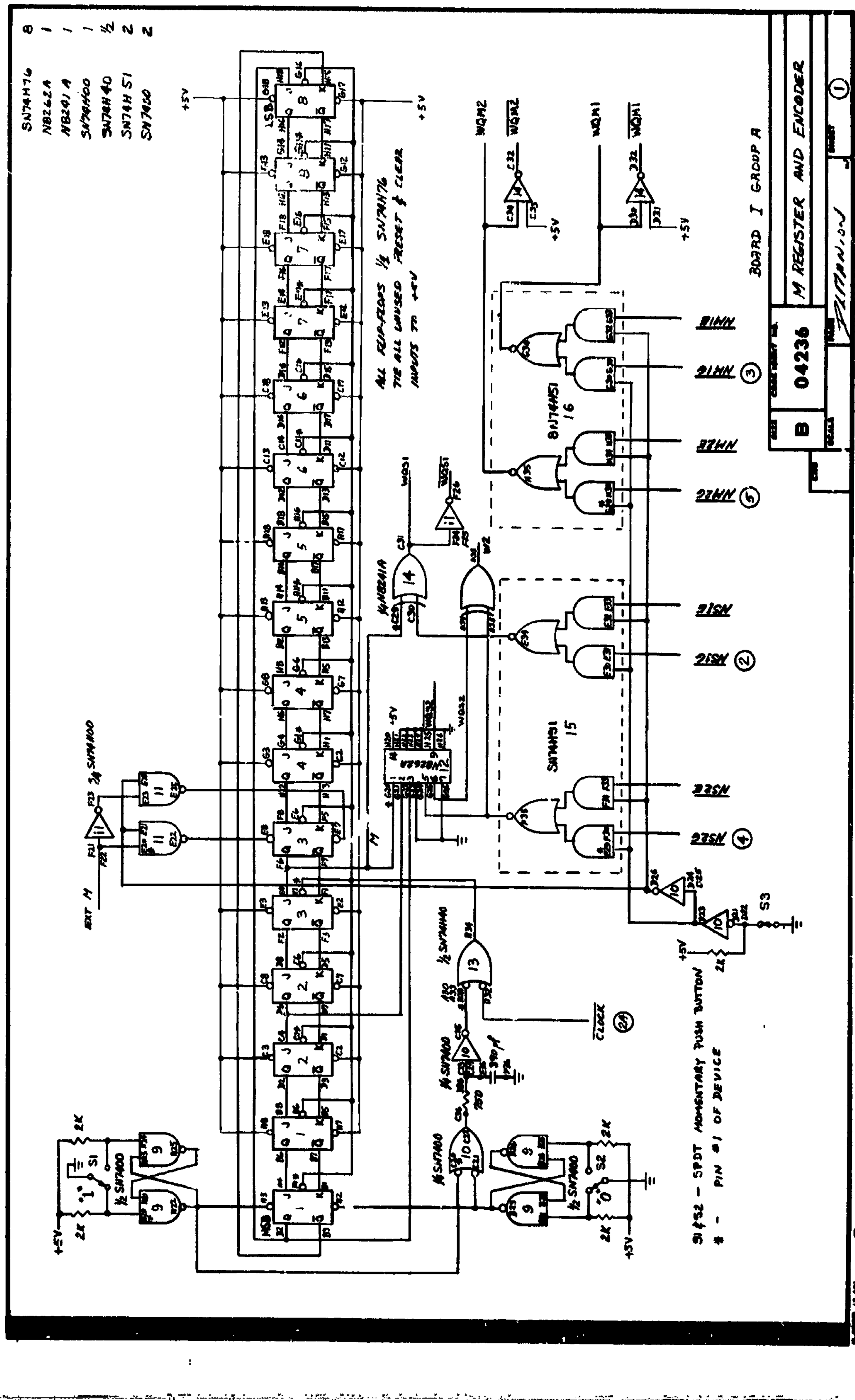

BOARD I GROUP A
M REGISTER AND ENCODER
B
04236
ALL FLIP-FLOPS 1/2 SN74H76
TIE ALL UNUSED PRESET & CLEAR INPUTS TO +5V
S1 & S2 - SPDT MOMENTARY PUSH BUTTON
- PIN #1 OF DEVICE
CLOCK
EXT M
LSB
MSB
+5V
SN74H76
NB262A
NB241A
SN74H00
SN74H40
SN74H51
SN7450
S3
WQM2
WQM1
WOS1
WOS2

Fig. V-2

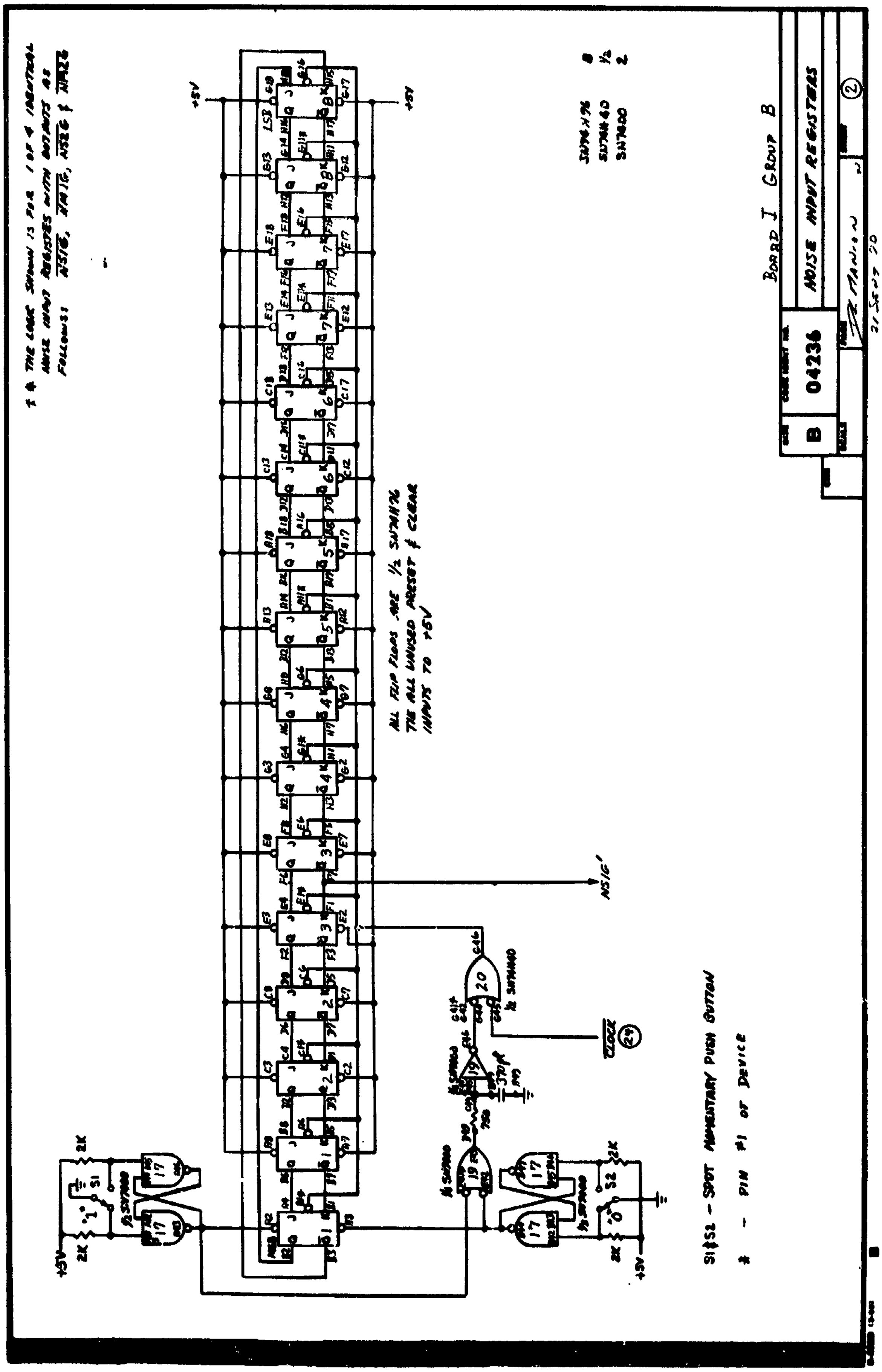

BOARD I GROUP B
NOISE INPUT REGISTERS
04236
ALL FLIP FLOPS ARE 1/2 SN7476
TIE ALL UNUSED PRESET & CLEAR
INPUTS TO +5V
S1 & S2 - SPOT MOMENTARY PUSH BUTTON
* - PIN #1 OF DEVICE
NSIG'
CLOCK
+5V
SN7476 SN7440 SN7400

Fig. V-3

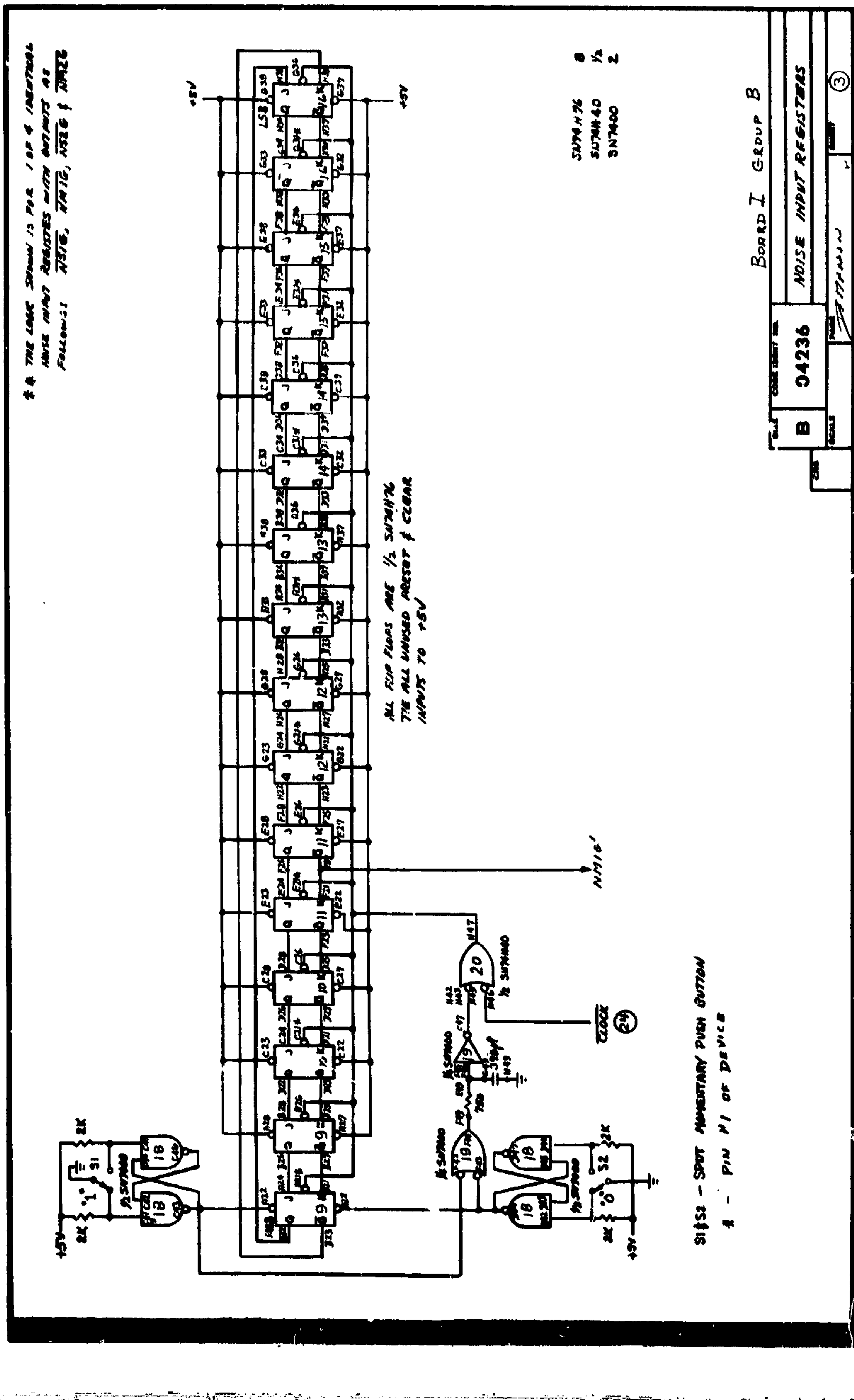


Fig. V-4

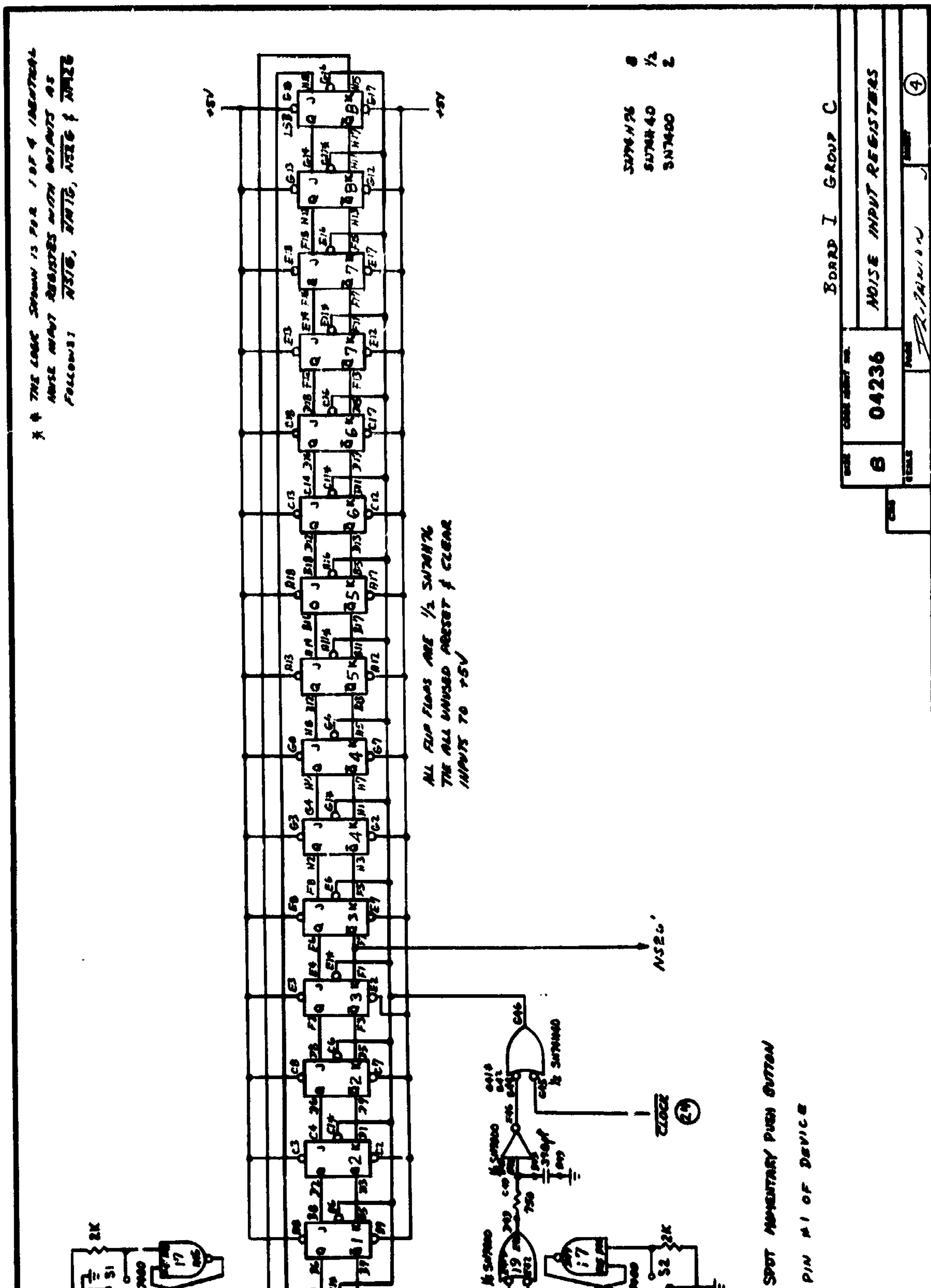

S1 & S2 – SPDT MOMENTARY PUSH BUTTON
* – PIN #1 OF DEVICE
ALL FLIP FLOPS ARE 1/2 SN7476
TIE ALL UNUSED PRESET & CLEAR
INPUTS TO +5V
NS2'
CLOCK
SN7476
SN7440
SN7400
BOARD I GROUP C
NOISE INPUT REGISTERS
B
04236
4

Fig. V-5

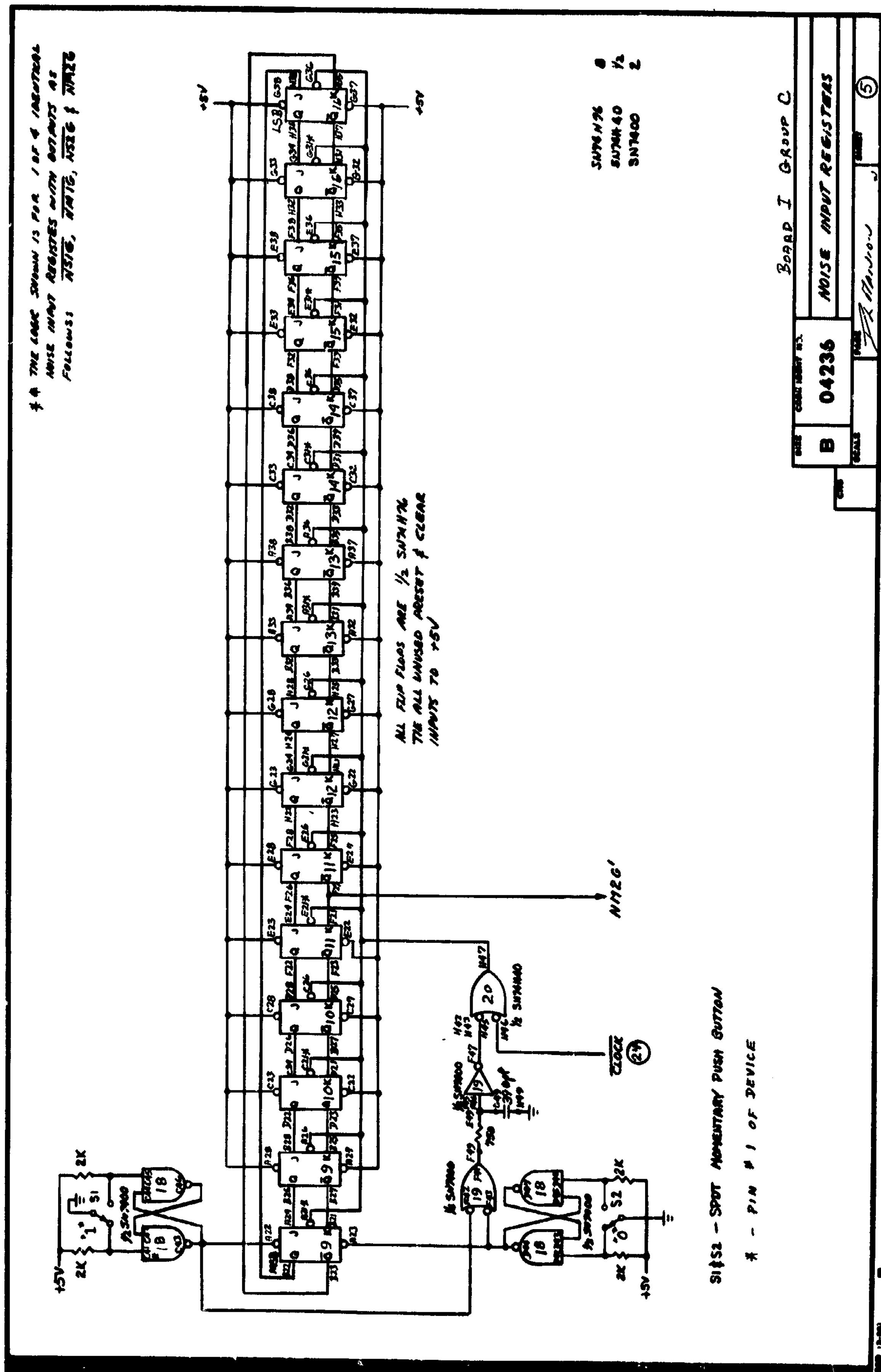

ALL FLIP FLOPS ARE 1/2 SN74H76
TIE ALL UNUSED PRESET & CLEAR
INPUTS TO +5V
S1 & S2 - SPDT MOMENTARY PUSH BUTTON
- PIN # 1 OF DEVICE
CLOCK
NM2G'
+5V
SN74H76
SN74H40
SN7400
BOARD I GROUP C
NOISE INPUT REGISTERS
B
04236

Fig. V-6

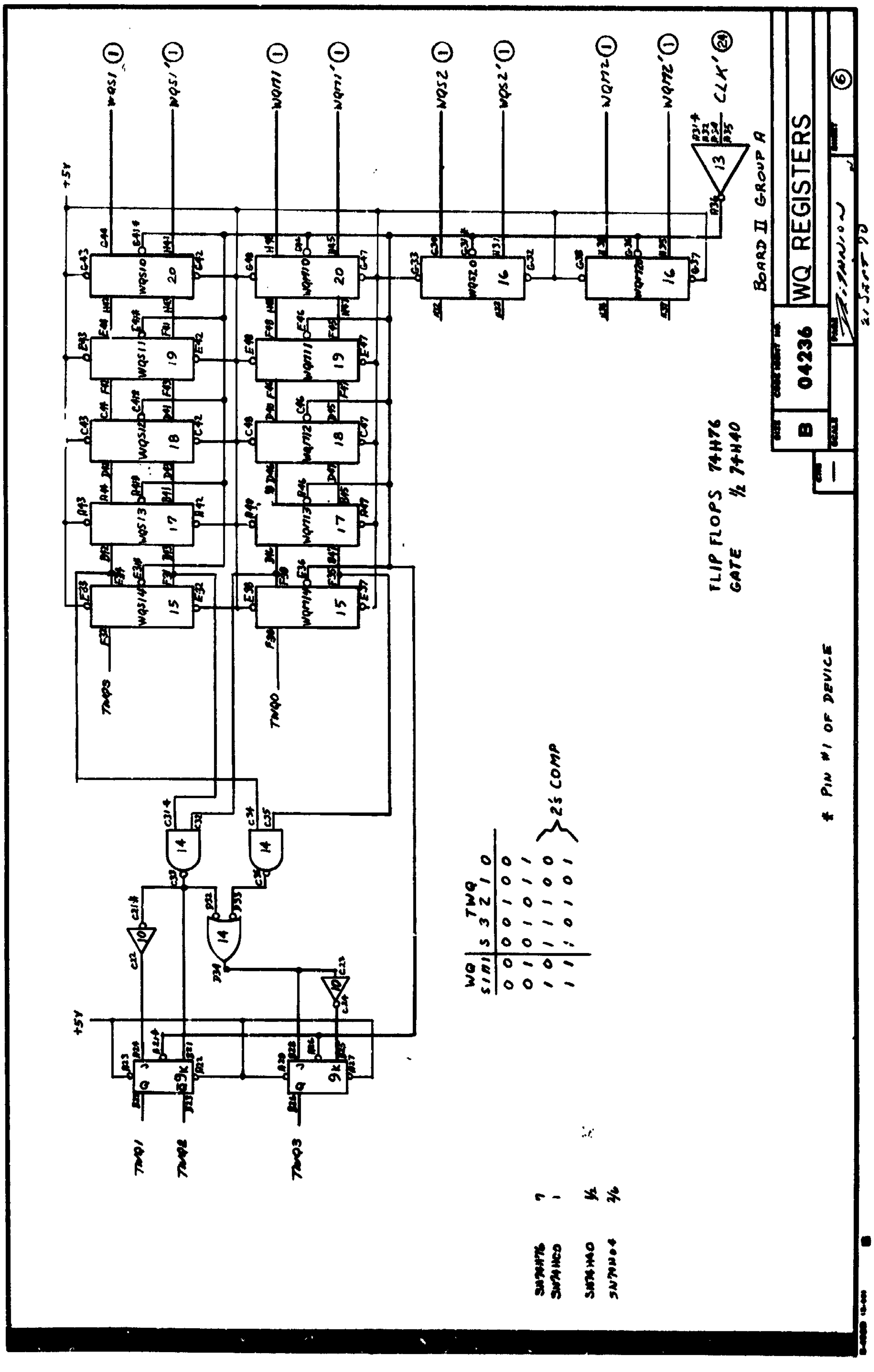

WQ REGISTERS
04236
B
BOARD II GROUP A
FLIP FLOPS 74H76
GATE 1/2 74H40
* PIN #1 OF DEVICE
2's COMP
CLK'
+5V
TWQ0
TWQ1
TWQ2
TWQ3

Fig. V-7

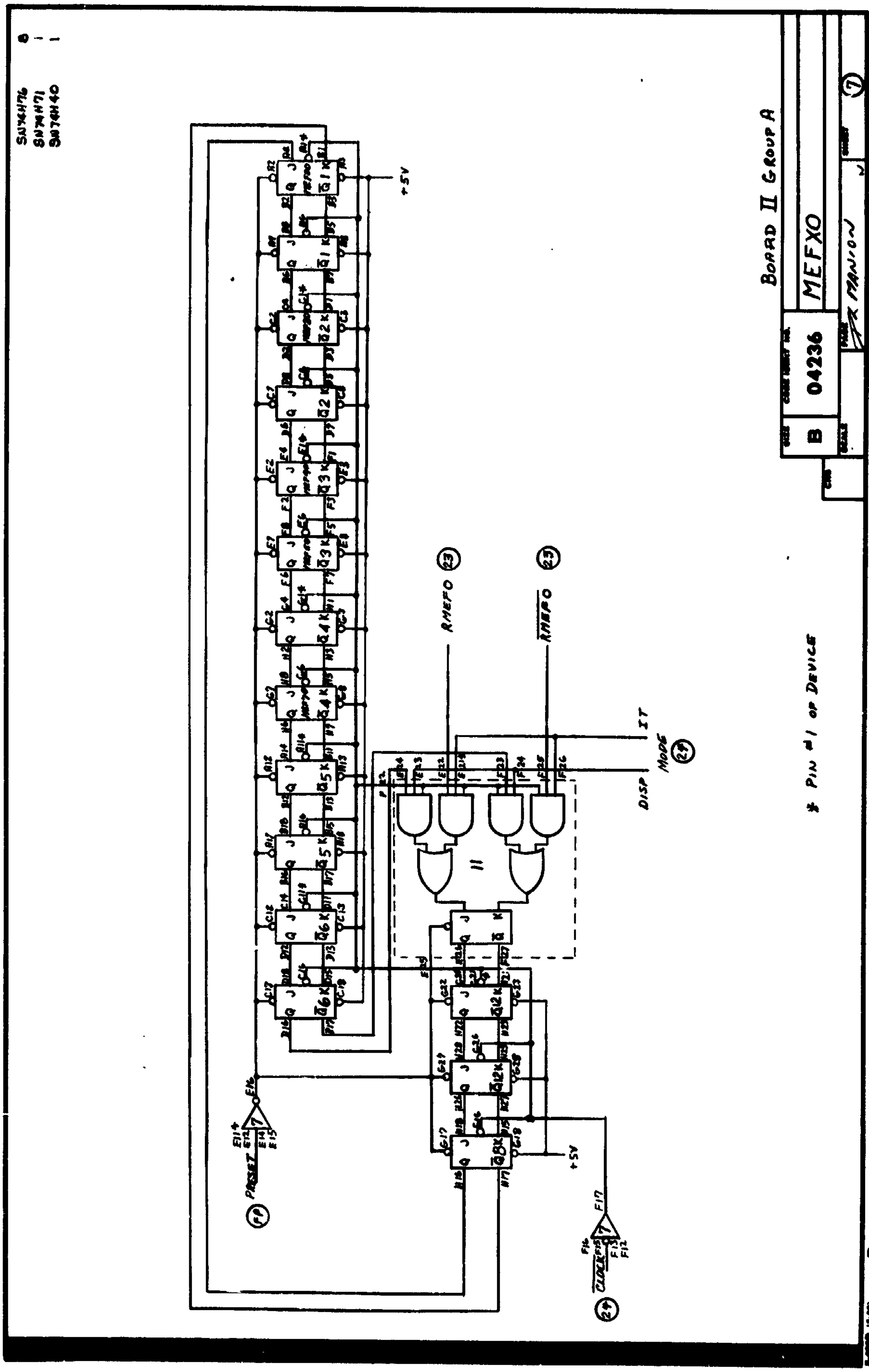

SN74H76
SN74H71
SN74H40
+5V
RMEFO
DISP
MODE
IT
PRESET
CLOCK
* Pin #1 of Device
Board II Group A
MEFXO
B
04236
18 Sept 70

Fig. V-8

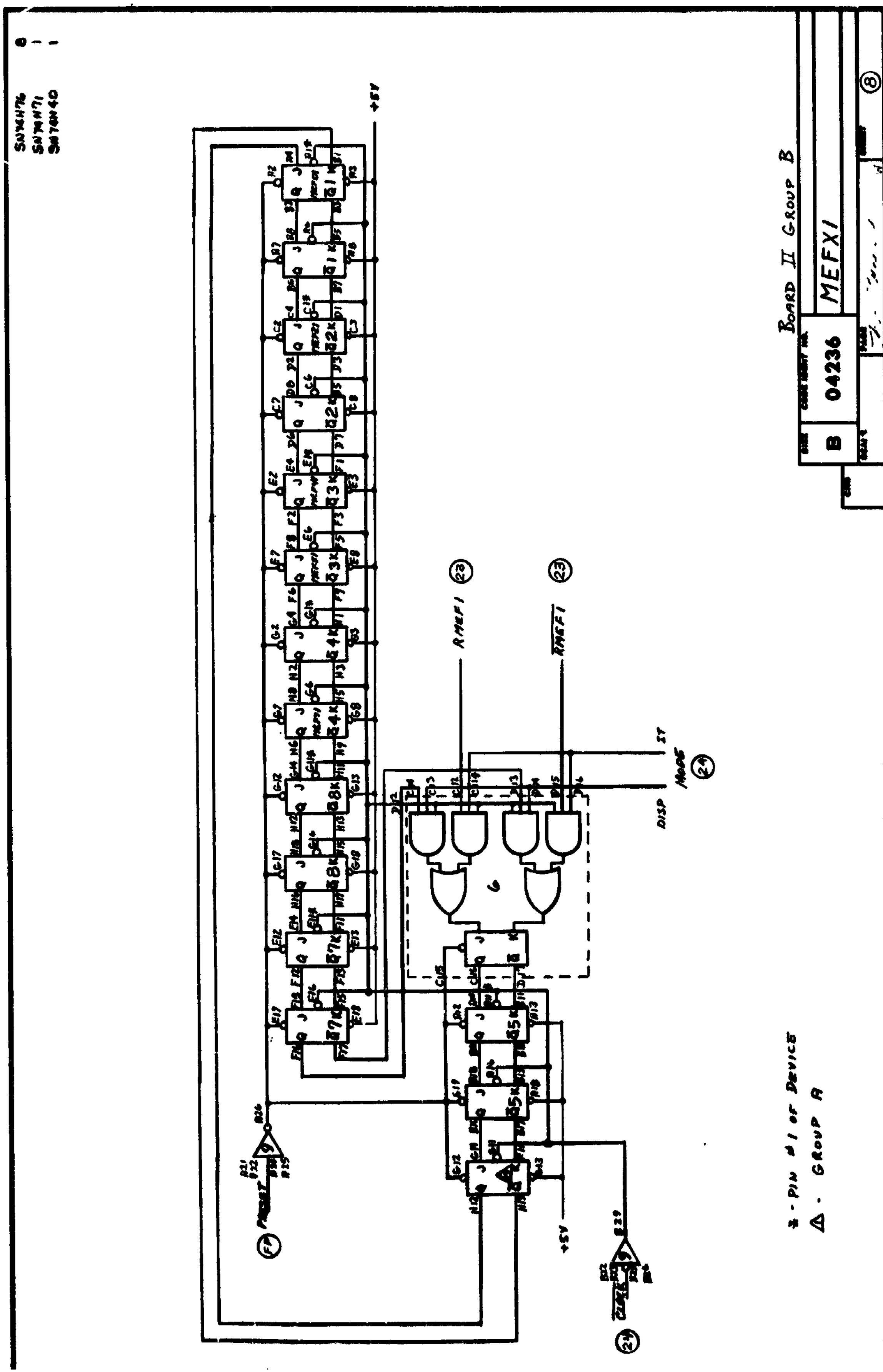

Board II Group B
MEFXI
B
04236
+5V
RMEF1
DISP
MODE
PRESET
CLEAR
* - PIN #1 OF DEVICE
Δ - GROUP A

Fig. V-9

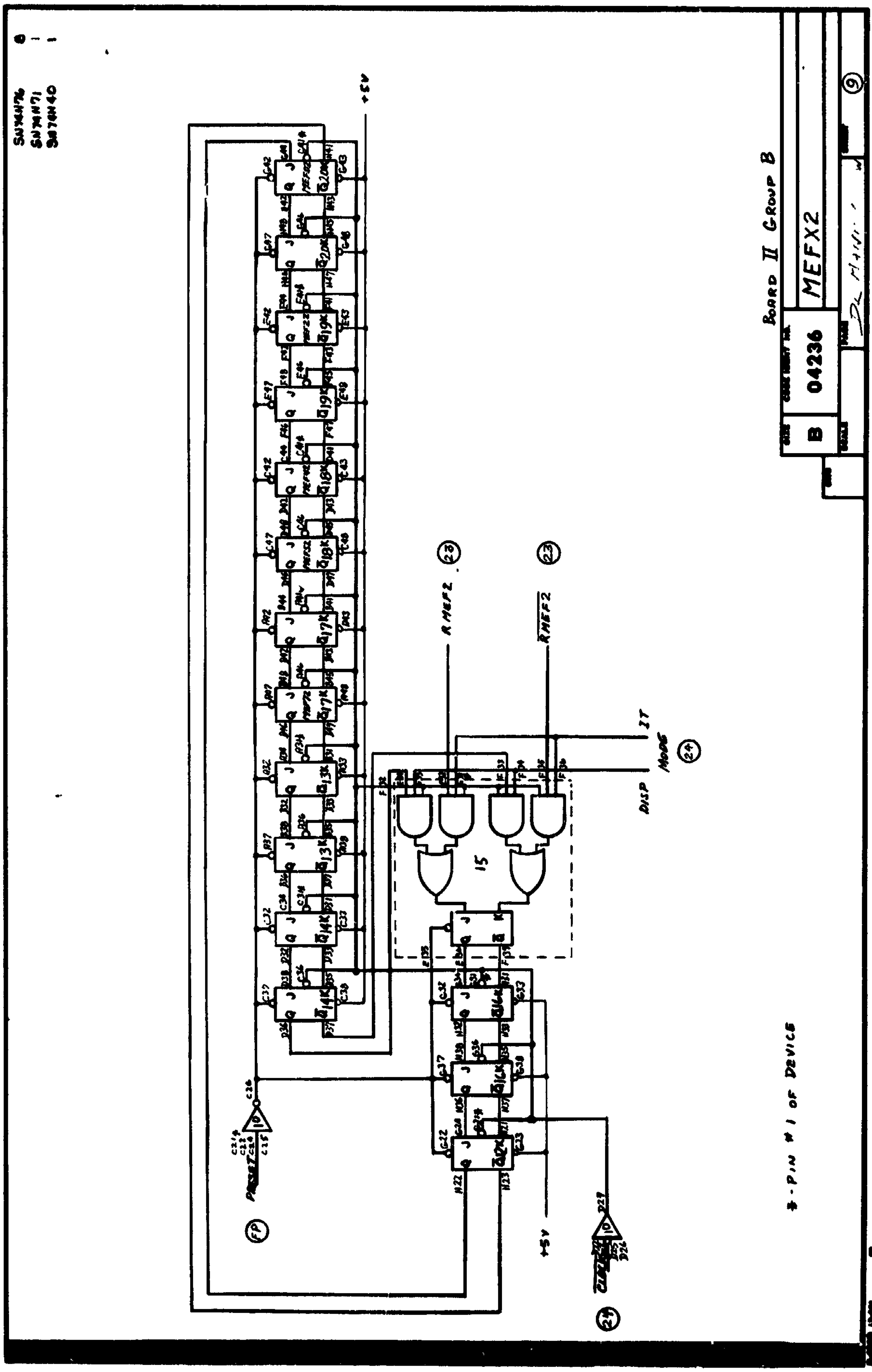

BOARD II GROUP B
MEFX2
B
04236
R MEF2
DISP
MODE
IT
PRESET
+5V
15
PIN # 1 OF DEVICE

Fig. V-10

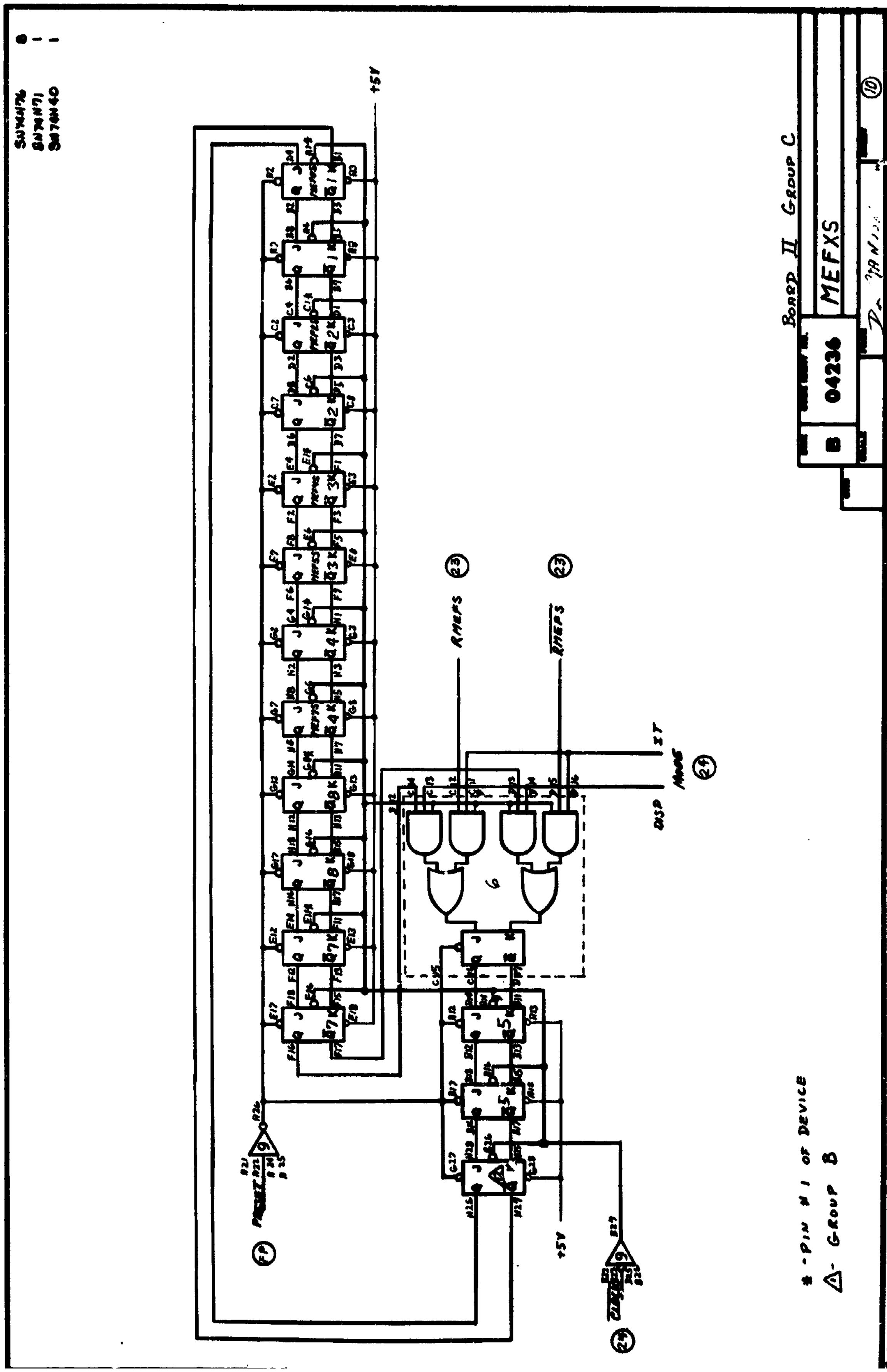

SN7476
SN74H71
SN74H40
+5V
RMEPS
23
DISP MODE
24
PRESET
FP
* - PIN # 1 OF DEVICE
Δ - GROUP B
BOARD II GROUP C
MEFXS
04236
B
22 SEPT 70

Fig. V-11

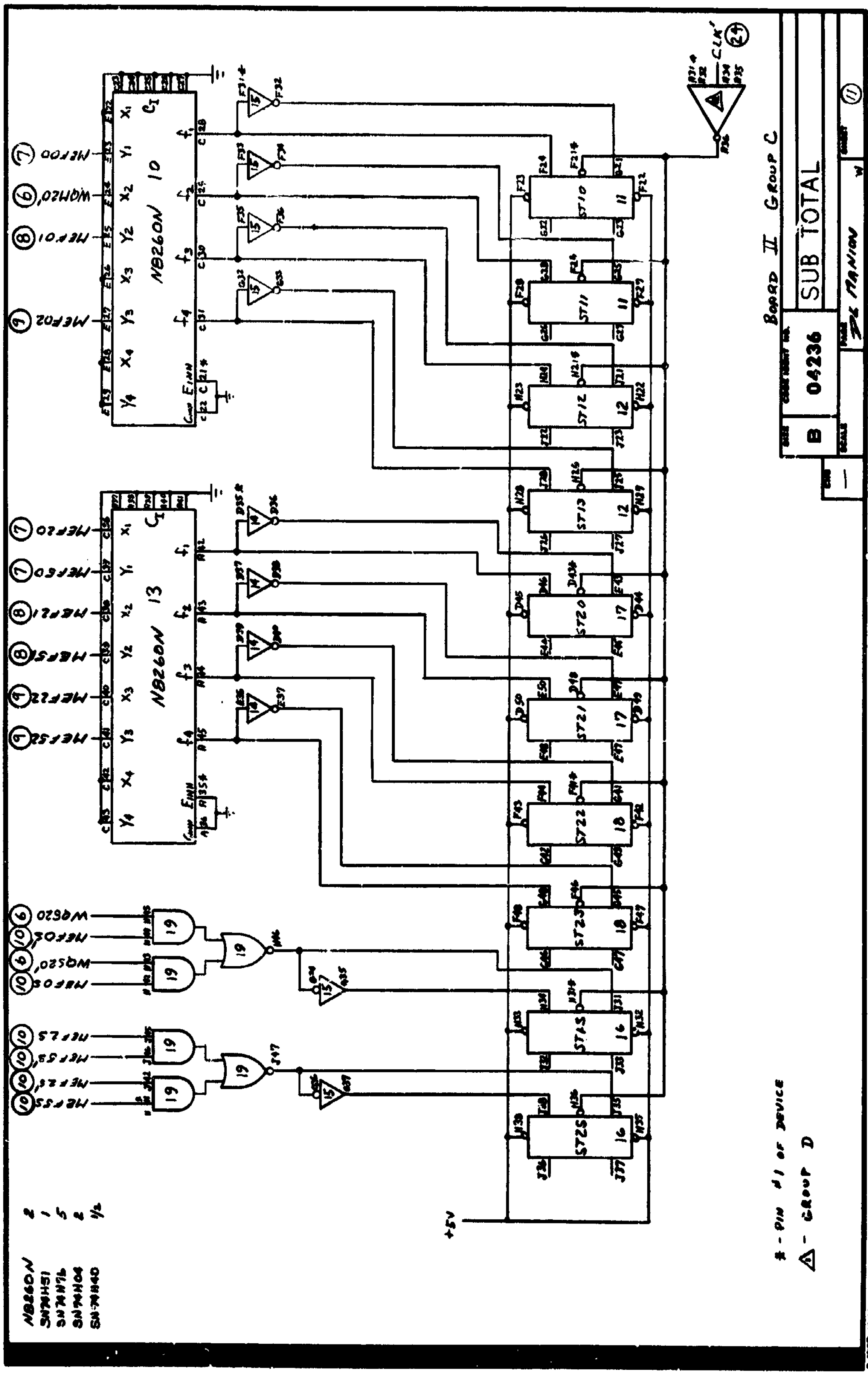

SUB TOTAL
BOARD II GROUP C
04236
27 AUG 70
NB260N 10
NB260N 13
NB260N
SN74H51
SN74H74
SN74H04
SN74H40
+5V
CLK'
- PIN #1 OF DEVICE
△ - GROUP D

Fig. V-12

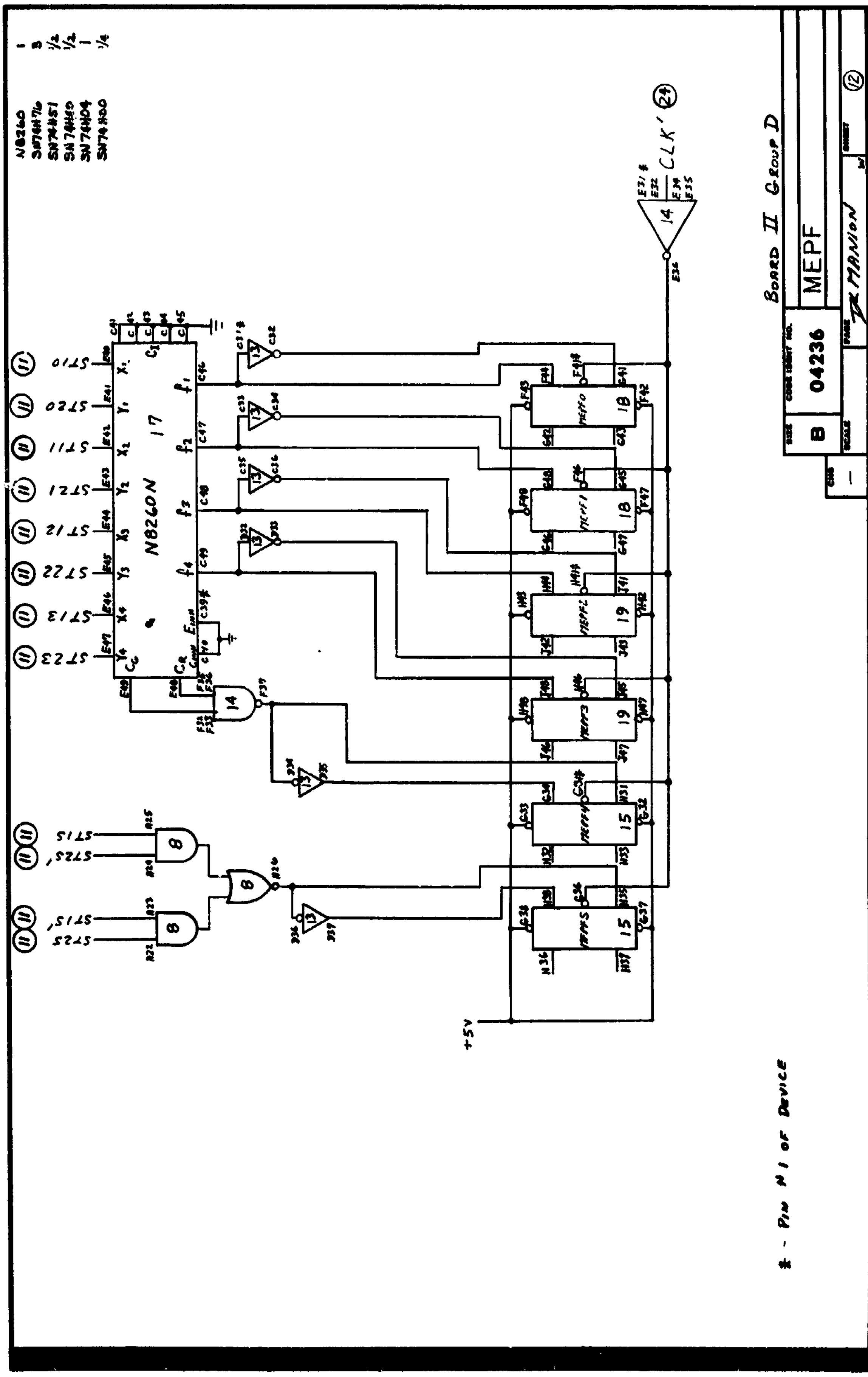

N8260N
17
CLK'
+5V
BOARD II GROUP D
MEPF
B
04236
29 AUG 70
* - PIN #1 OF DEVICE

Fig. V-13

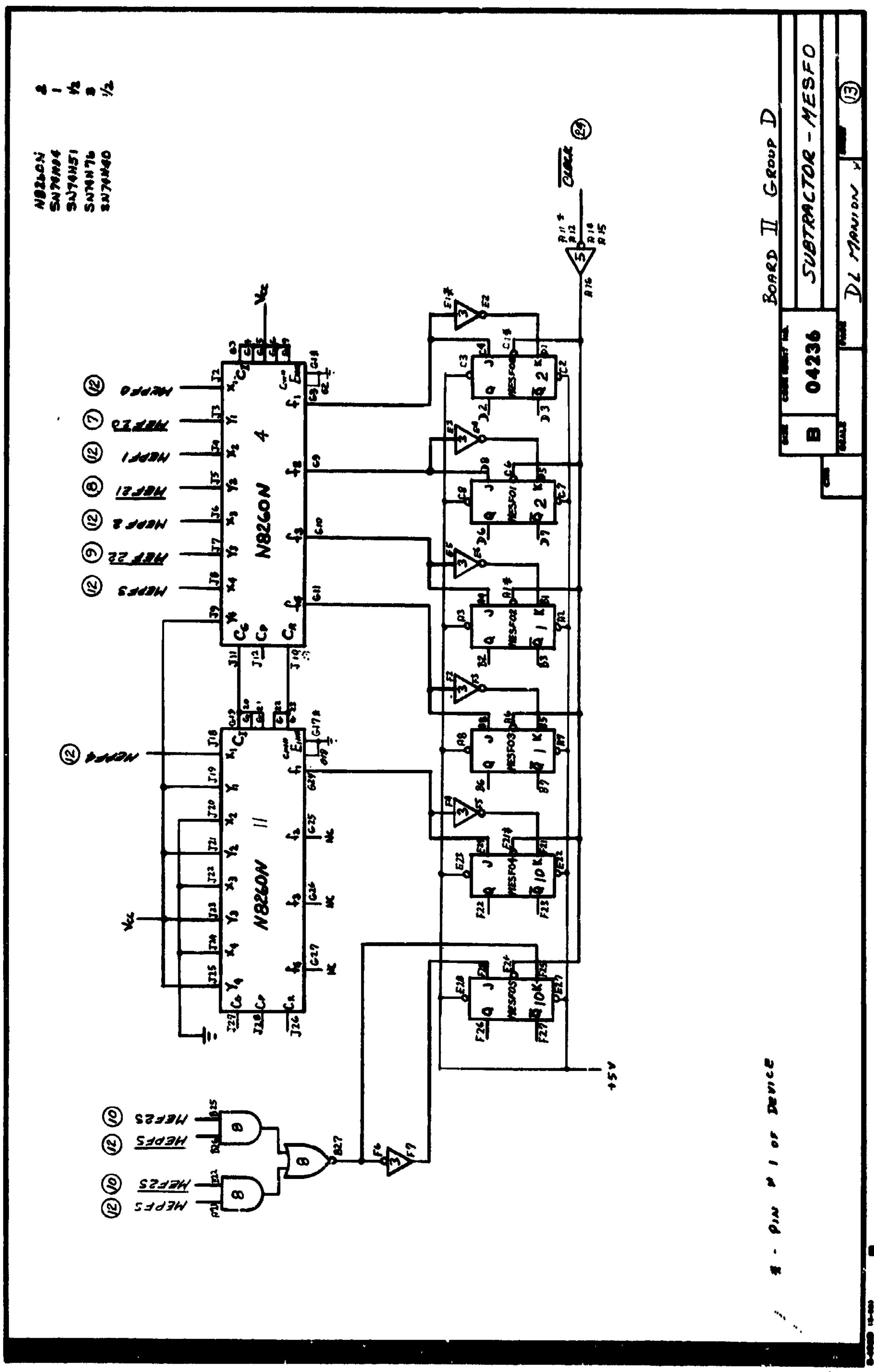

Board II Group D
SUBTRACTOR - MESFO
04236
DL Manion
25 Sept 70
N8260N
+5V

Fig. V-14

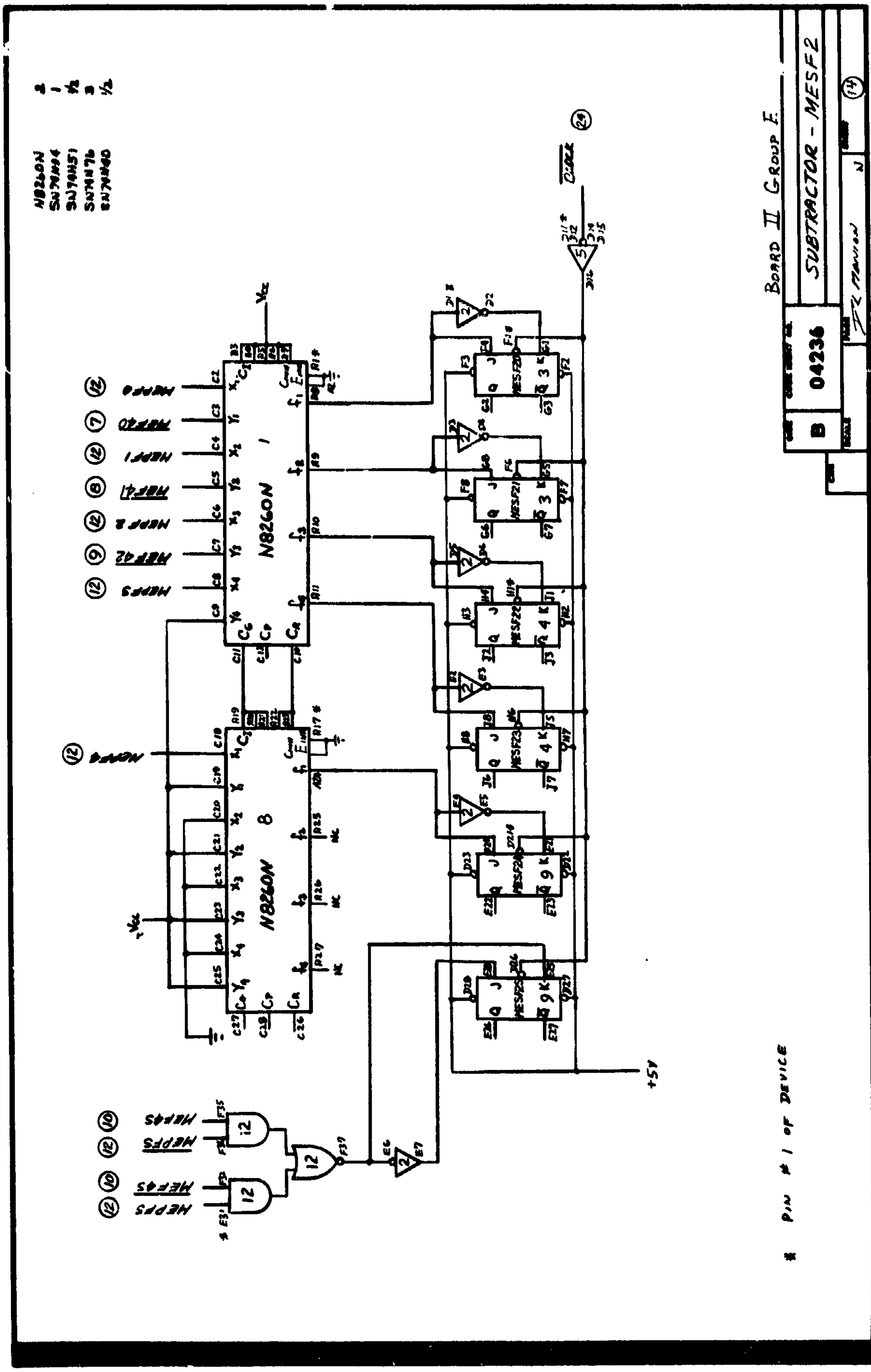

N8260N
SN7404
SN74H51
SN7476
SN7440
BOARD II GROUP F
SUBTRACTOR - MESF2
B
04236
+5V
PIN # 1 OF DEVICE

Fig. V-15

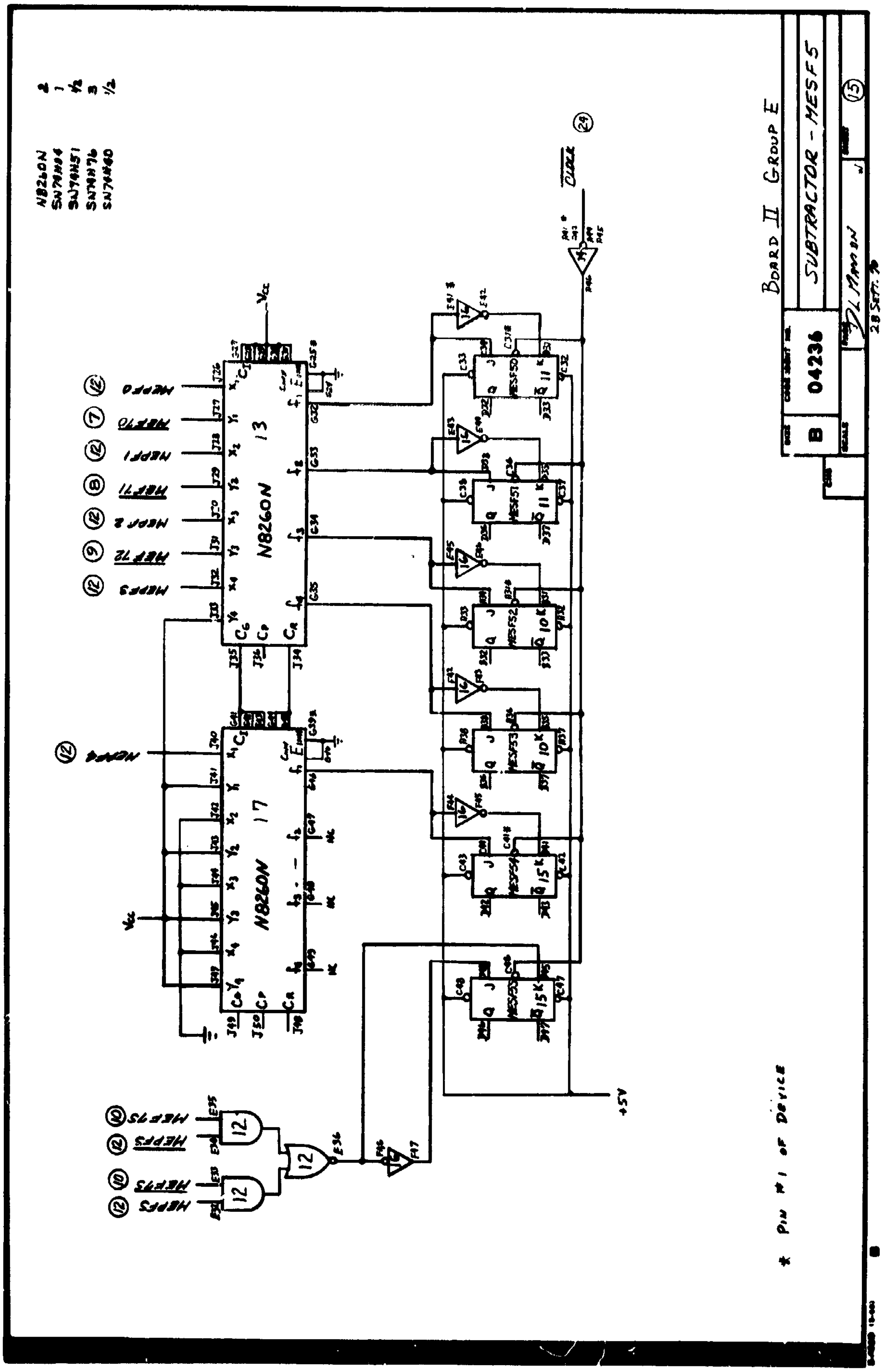

BOARD II GROUP E
SUBTRACTOR - MESF5
04236
N8260N
+5V
* PIN #1 OF DEVICE

Fig. V-16

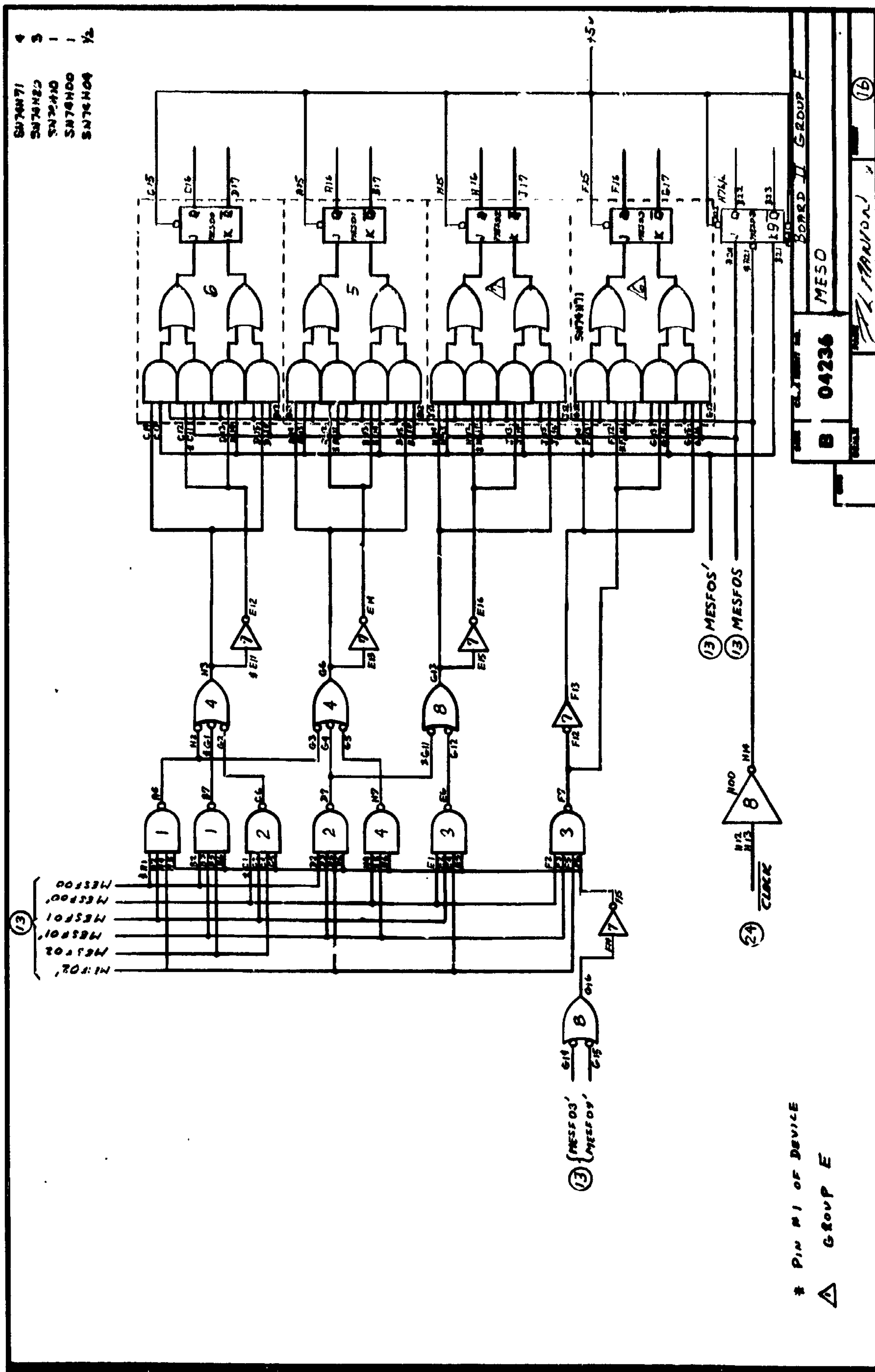

MESFOS'
MESFOS
CLOCK
+5v
BOARD II GROUP F
MESO
04236
B
* PIN #1 OF DEVICE
GROUP E

Fig. V-17

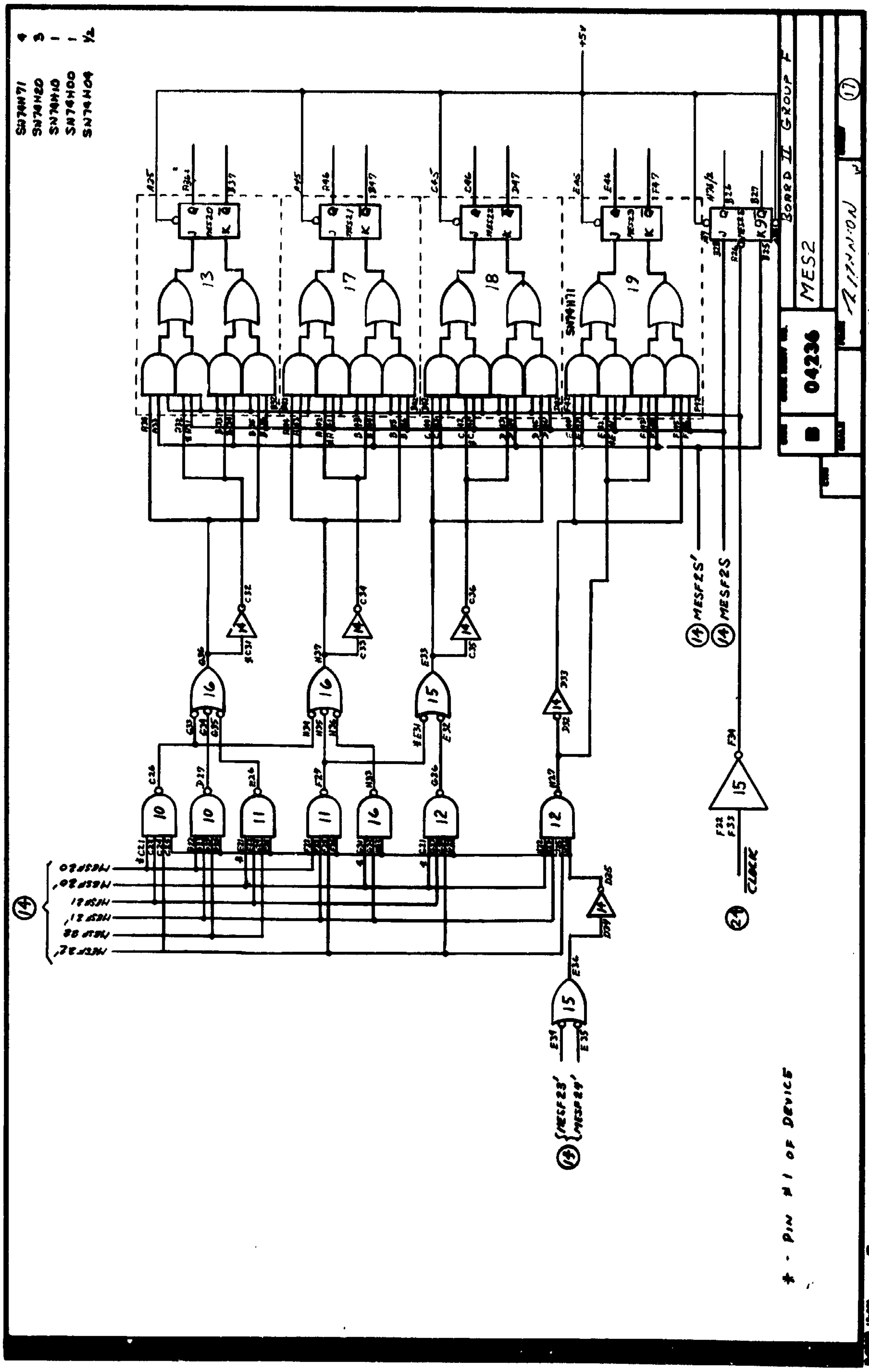

SN74H71
SN74H20
SN74H10
SN74H00
SN74H04
BOARD II GROUP F
MES2
04236
MESF2S'
MESF2S
CLEAR
* - PIN #1 OF DEVICE

Fig. V-18

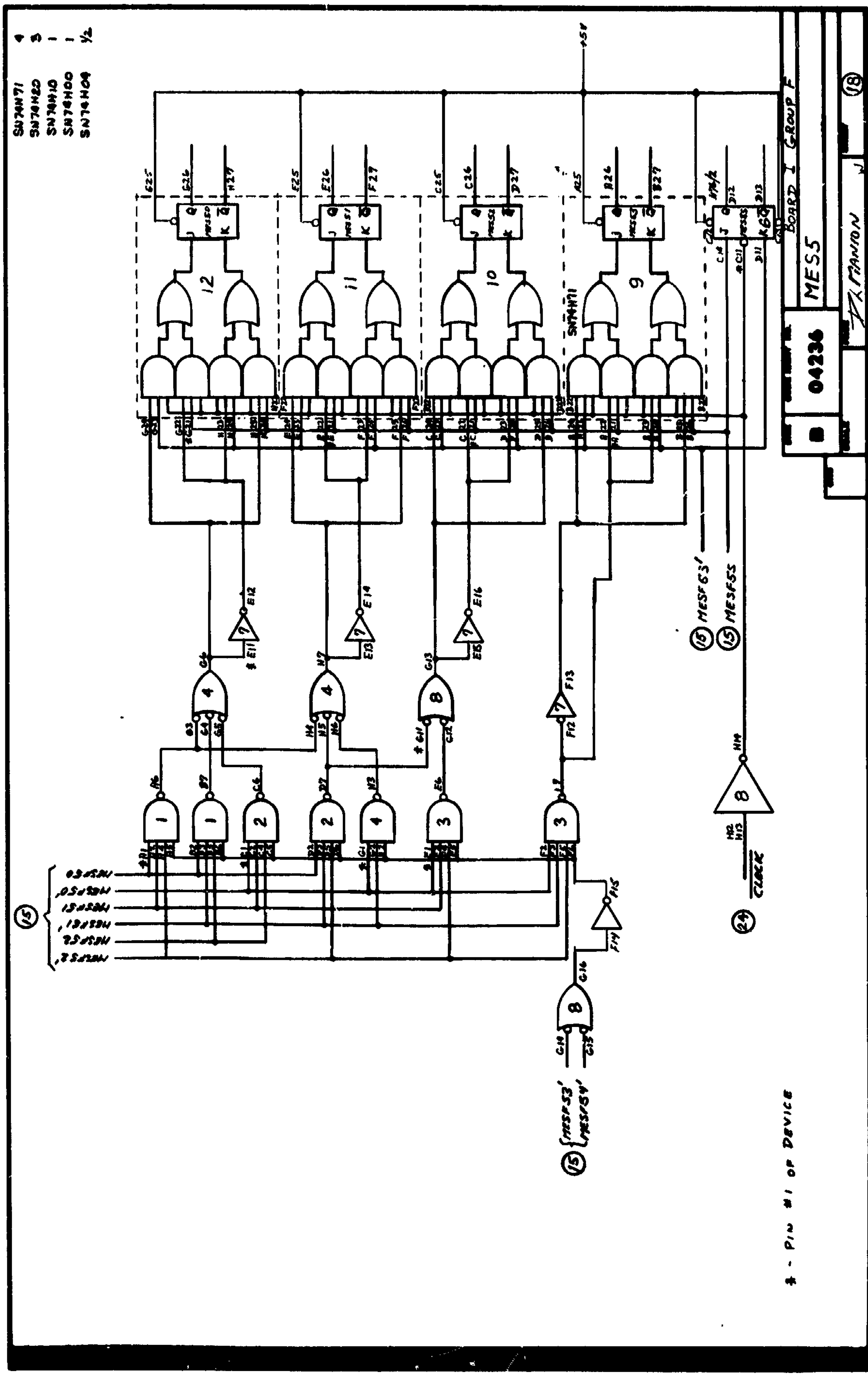

MESS
04236
BOARD I GROUP F
21 SEPT 70
$ - PIN #1 OF DEVICE
CLOCK
MESF63'
MESF55
+5V

Fig. V-19

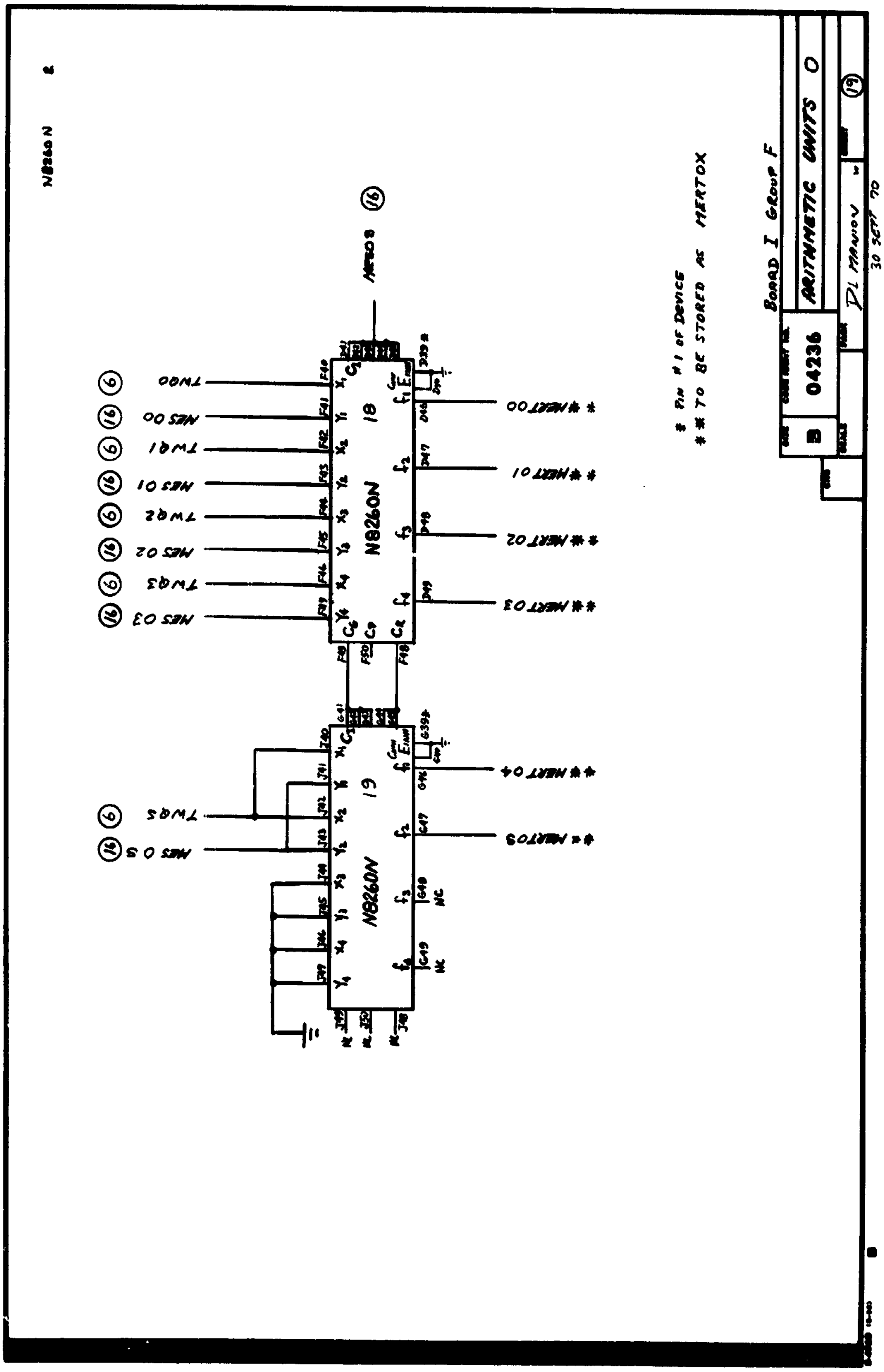

N8260N
TW00
NES 00
TW01
NES 01
TW02
NES 02
TW03
NES 03
TW05
NES 05
MERO8
N8260N
18
N8260N
19
** MERT00
** MERT01
** MERT02
** MERT03
** MERT04
** MERT05
# Pin # 1 of Device
** To be stored as MERTOX
Board I Group F
ARITHMETIC UNITS
04236
D L Marion
30 SEPT 70

Fig. V-20

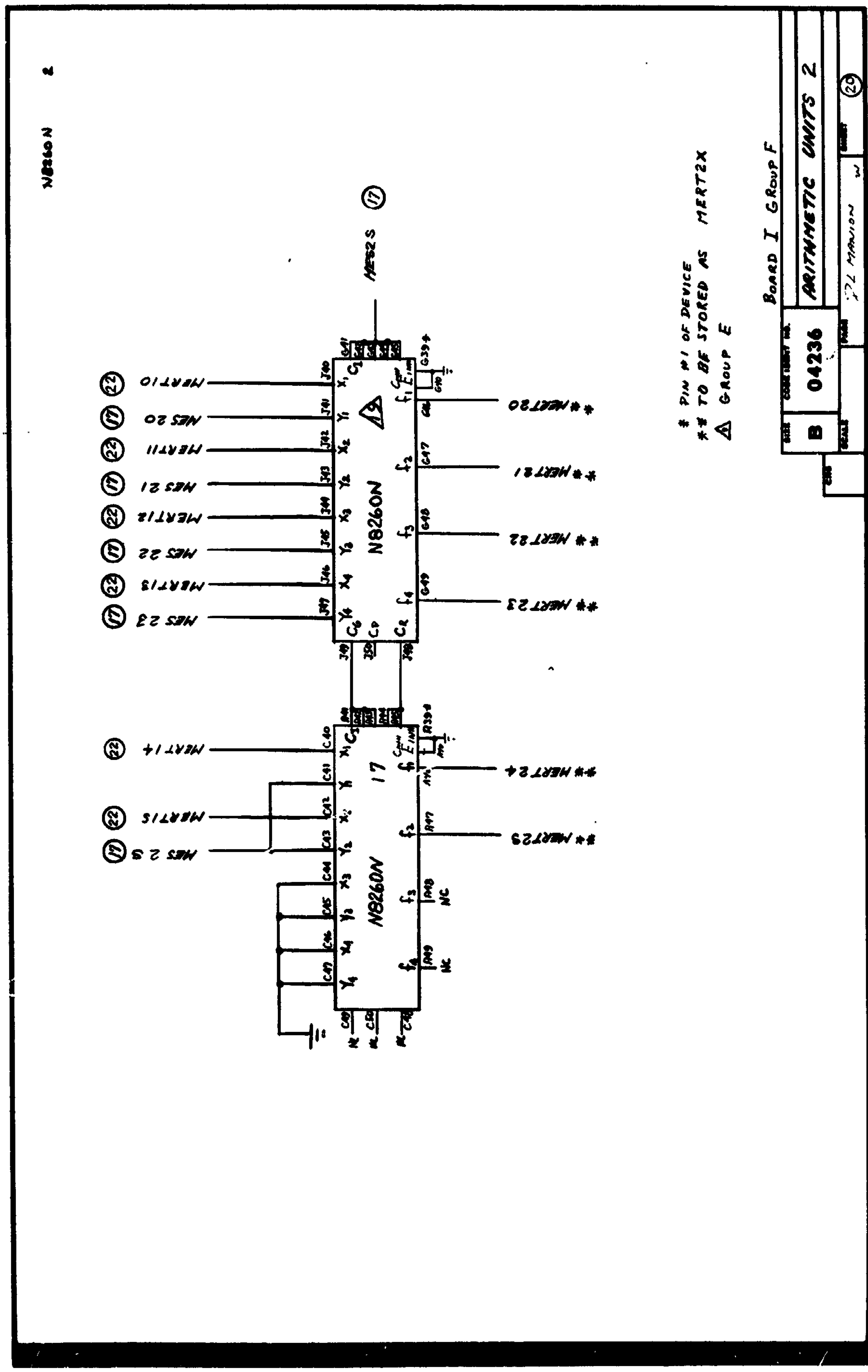

N8260N
MERT10
MES 20
MERT11
MES 21
MERT12
MES 22
MERT13
MES 23
MES25
MERT20
MERT21
MERT22
MERT23
MERT14
MERT15
MES 25
MERT24
MERT25
17
N8260N
PIN #1 OF DEVICE
TO BE STORED AS MERT2X
Δ GROUP E
BOARD I GROUP F
ARITHMETIC UNITS 2
B
04236
30 SEPT. 70

Fig. V-21

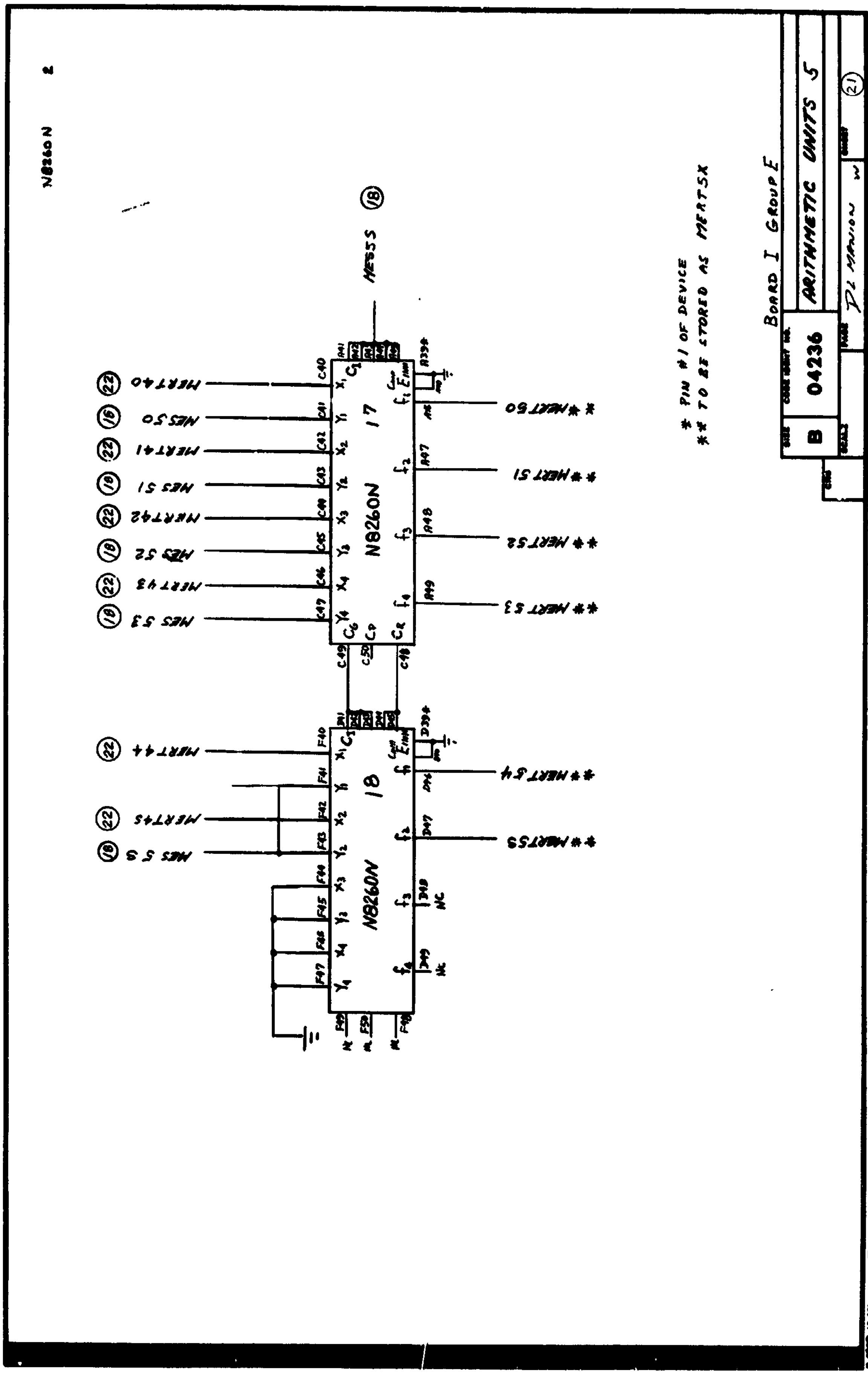

N8260N
17
18
MERT40
MES50
MERT41
MES51
MERT42
MES52
MERT43
MES53
MERT44
MERT45
MES55
MES55
** MERT50
** MERT51
** MERT52
** MERT53
** MERT54
** MERT55
* PIN #1 OF DEVICE
** TO BE STORED AS MERTSX
BOARD I GROUP E
ARITHMETIC UNITS 5
B
04236
30 SEPT 70

Fig. V-22

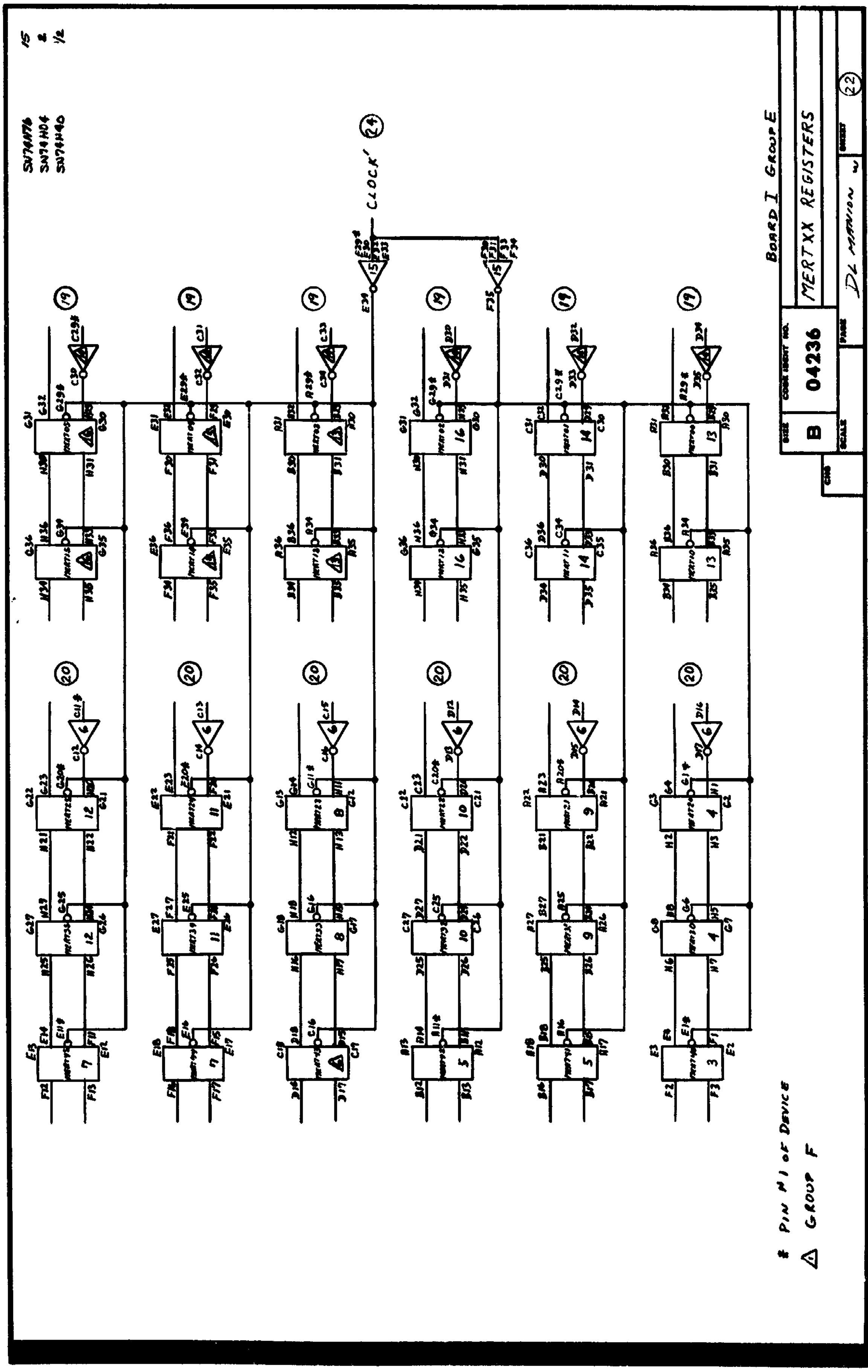

SN74H76 15
SN74H04 6
SN74H40 1/2
CLOCK' 24
BOARD I GROUP E
MERTXX REGISTERS
DL MANION
B
04236
22
* PIN #1 OF DEVICE
GROUP F

Fig. V-23

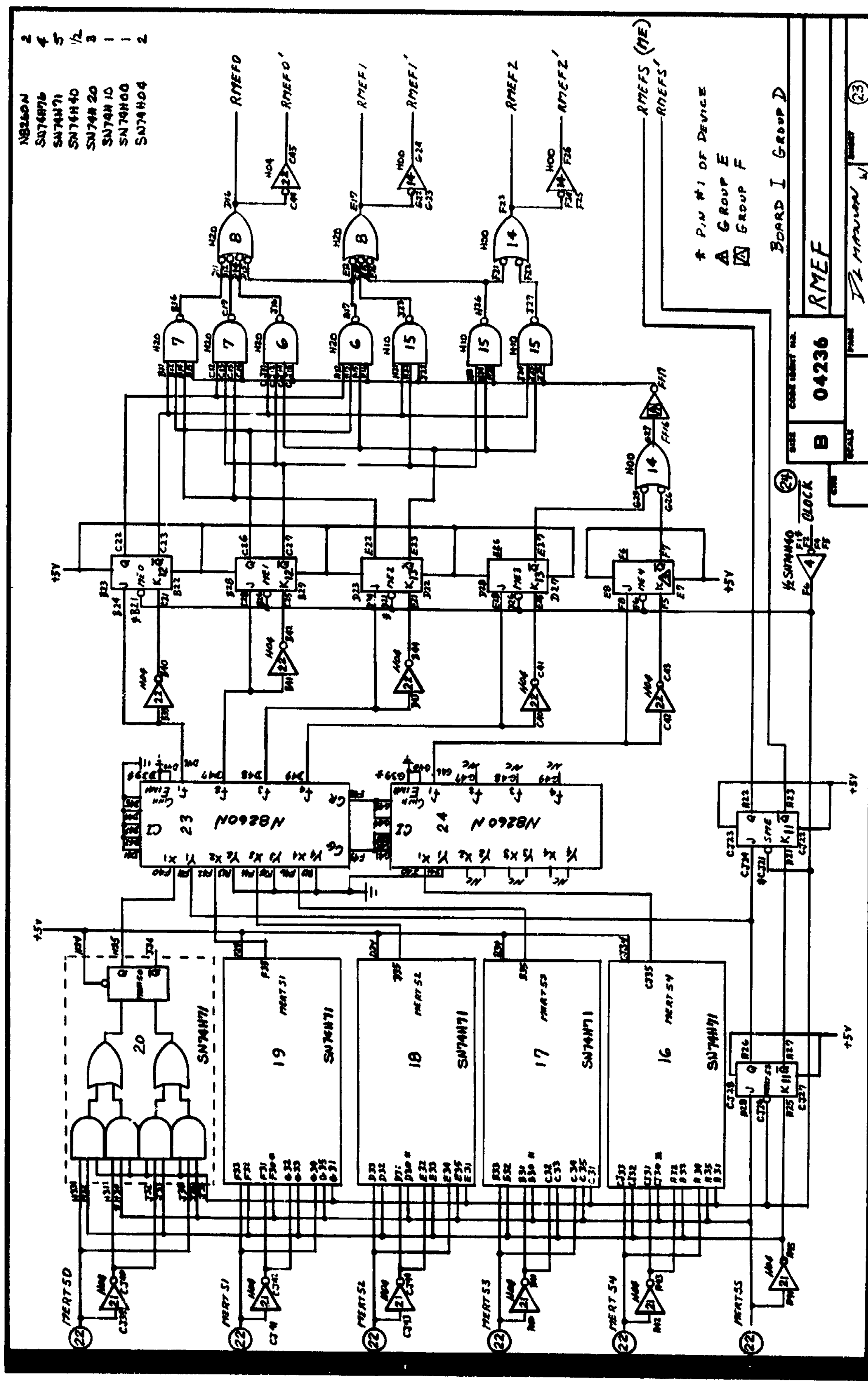

RMEF0
RMEF0'
RMEF1
RMEF1'
RMEF2
RMEF2'
RMEFS (ME)
RMEFS'
CLOCK
N8260N
SN74H71
BOARD I GROUP D
RMEF
04236
24 SEPT 70

Fig. V-24

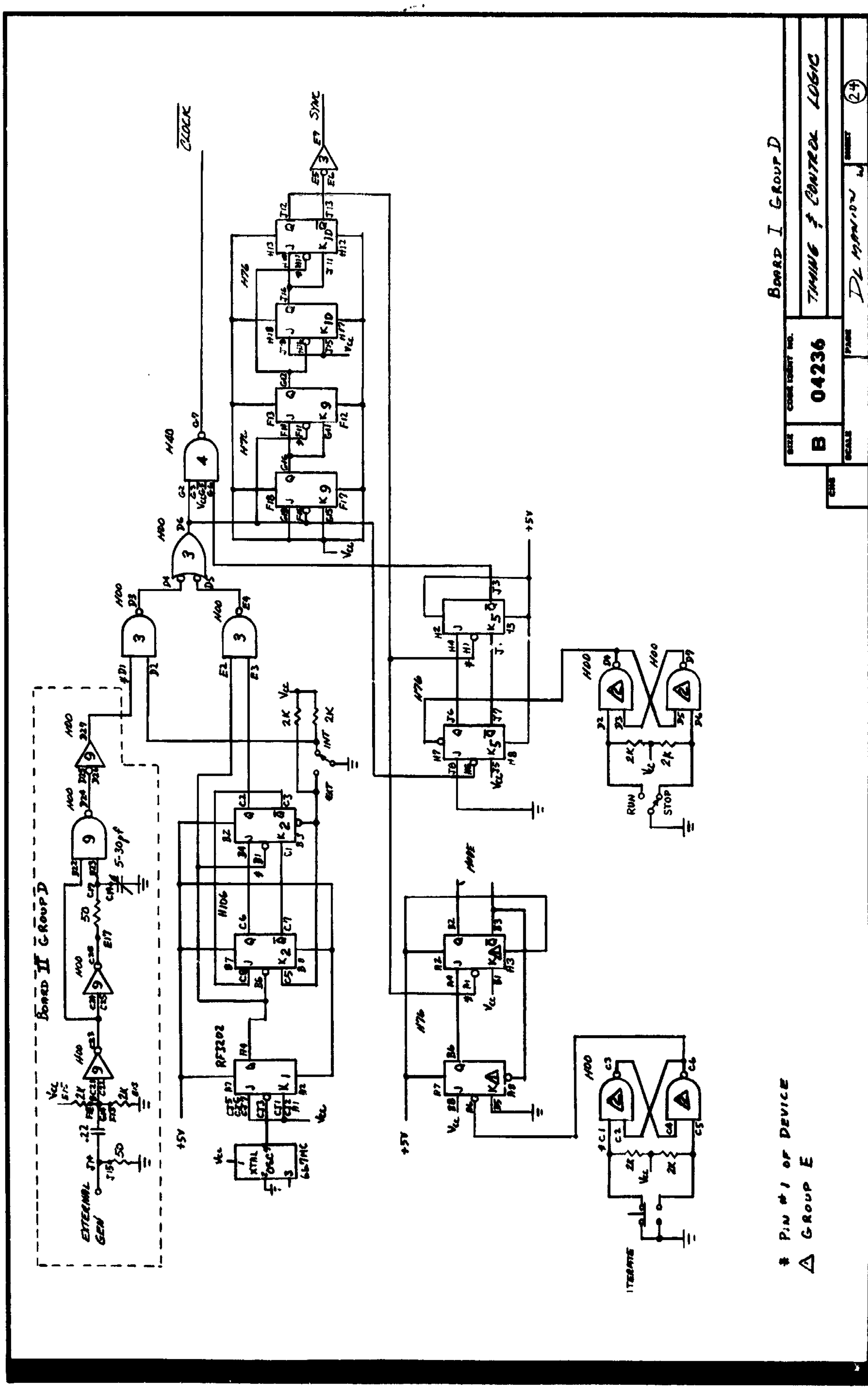

Board I Group D
Timing & Control Logic
B
04236
16 Sept 1970
CLOCK
SYNC
Board II Group D
External Gen
RUN
STOP
ITERATE
MODE
INT
EXT
+5V
Pin #1 of Device
Δ Group E

Fig. V-25

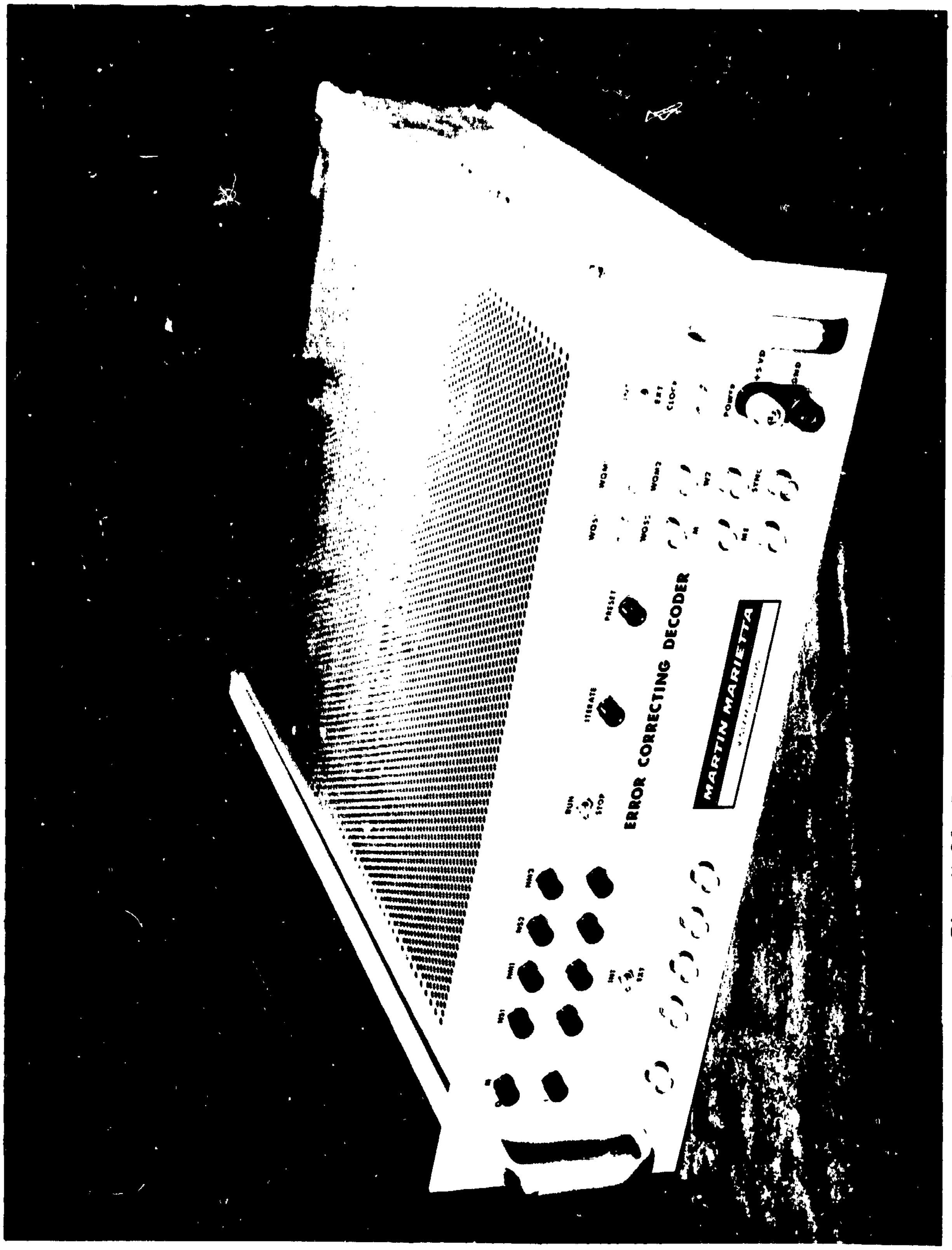


Fig. V-26 Demonstration Decoder Front View (Closed)

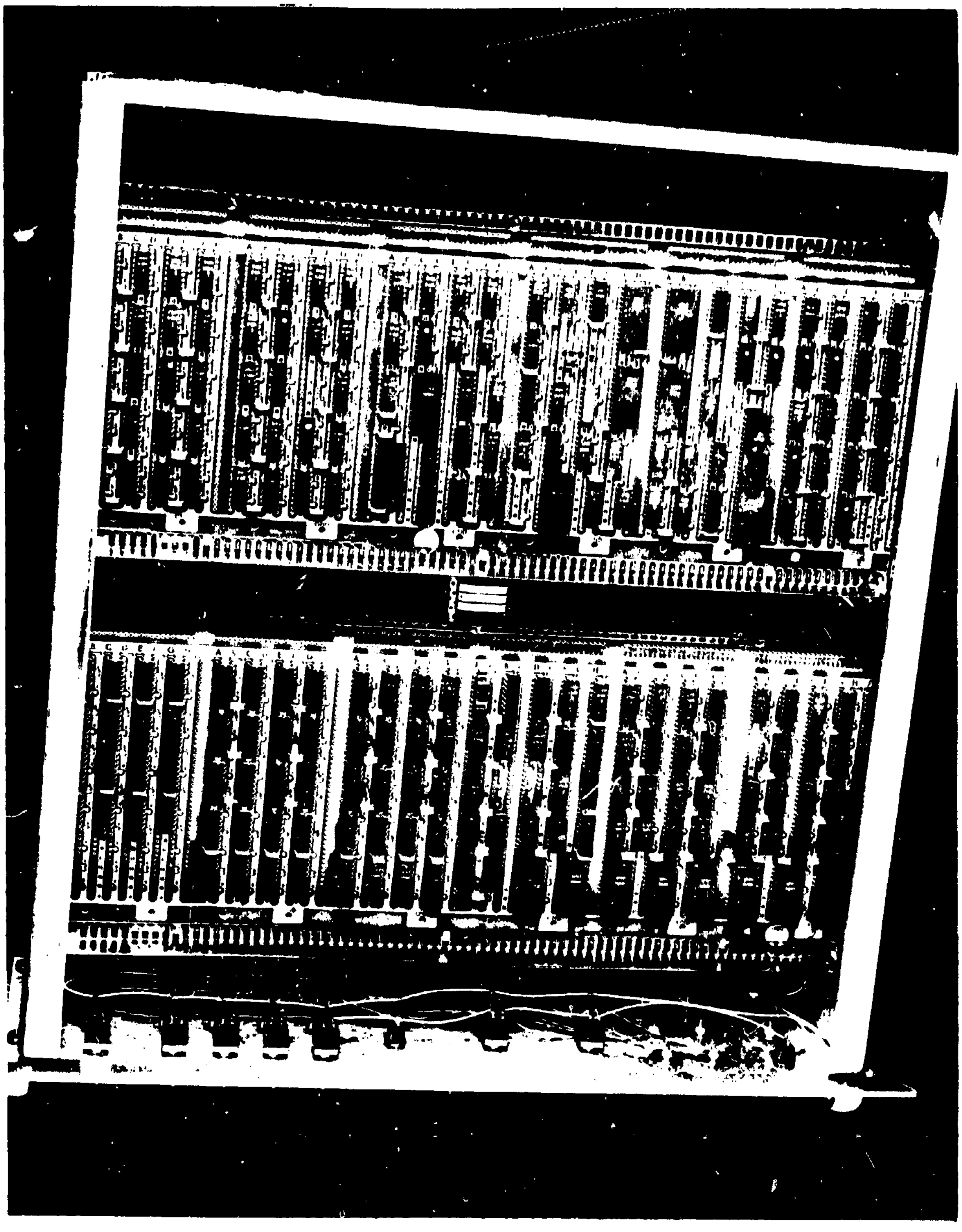


Fig. V-27 Demonstration Decoder Top View (Inside)

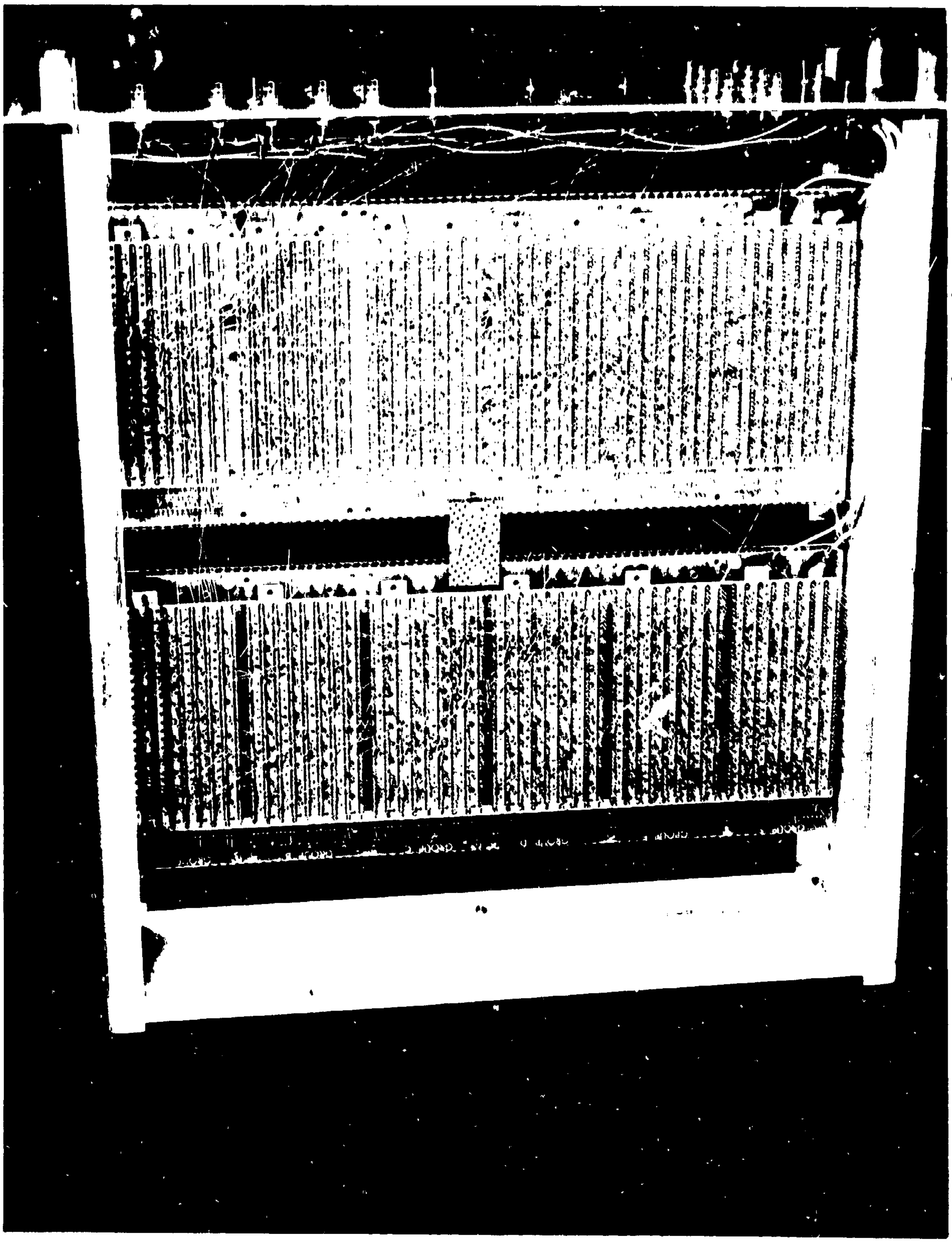


Fig. V-28 Demonstration Decoder Bottom View (Inside)

VI. CONVOLUTIONAL CODE PERFORMANCE

This chapter contains tables of convolutional codes suitable for use with the Viterbi decoding algorithm. Good codes are listed for constraint lengths K from 4 to 9 and rate denominators V from 2 to 4. Performance figures are also given, although caution must be used, as explained later in this report, in using these figures at low signal-to-noise ratio and large constraint length.

When investigation of Viterbi decoding was started, it was found that selection and evaluation of codes presented some difficulties that are alleviated by these tables. Two criteria must be met for good codes -- they must not be "catastrophic" and they should have small numbers of code words of small weight. A pencil and paper testing of codes that meet these criteria is not difficult for constraint length K = 4 or 5, and V = 2 or 3, but it rapidly becomes tedious and uneconomic for larger K and V because of the astronomical number of codes that must be tried and the likelihood of human error.

Performance of the codes can be determined by simulating the decoding algorithm on a general purpose computer, but this Monte Carlo method requires a large amount of computer time to get statistically valid data at high signal-to-noise ratio or large K. A useful approximation to error frequency, the union bound, is valid at high signal-to-noise and is easily computed from a list of code word weights.

Our computer program calculates the union bound at energy signal-to-noise ratio per bit E_b/N_o = 4 dB and 6 dB for both bit and burst error frequencies, and lists the 10 best codes for each of the four criteria (4 dB and 6 dB, bits and bursts) for V = 2 to 4 and K = 4 to 9.

A. CONVOLUTIONAL CODES

Convolutional codes are easily generated with an encoder as shown in Fig. VI-1. A stream of message bits is shifted into the K-stage shift register one at a time. A number V of mod-2 adders (parity check circuits) are connected to various stages of the shift register and are each sampled by a commutator every time a new bit is shifted in. The coded output thus consists of V times as many channel symbols as message bits, and the *rate* of the code is said to be 1/V. A finite sequence of message bits is called a code word.

The Hamming distance (or simply "distance") between any two code words is the number of places in which they differ. Thus, 100111 and 110101 differ in two places, the 2nd and 5th, and therefore have a distance of 2. If the distance between two code words A and B is small, it is easy for channel noise to make the receiver think that B was transmitted when in fact it was A, and vice versa. In other words, A plus the noise can easily look more like B than like A.

Thus a code with a large minimum distance between code words is desirable. It is a fundamental property of parity check codes, including convolutional codes, that a list of the Hamming distances between a particular code word A and all other words in the code are independent of A. This simplifies the problem of finding the minimum distance since we can examine the distances from the all-zero code word 000...00. The Hamming distance from this word to a second word is simply the number of ones in the second word, and this number is called the Hamming weight (or simply the "weight") of the word. It is much easier to test a given code for the minimum weight than for minimum distance.

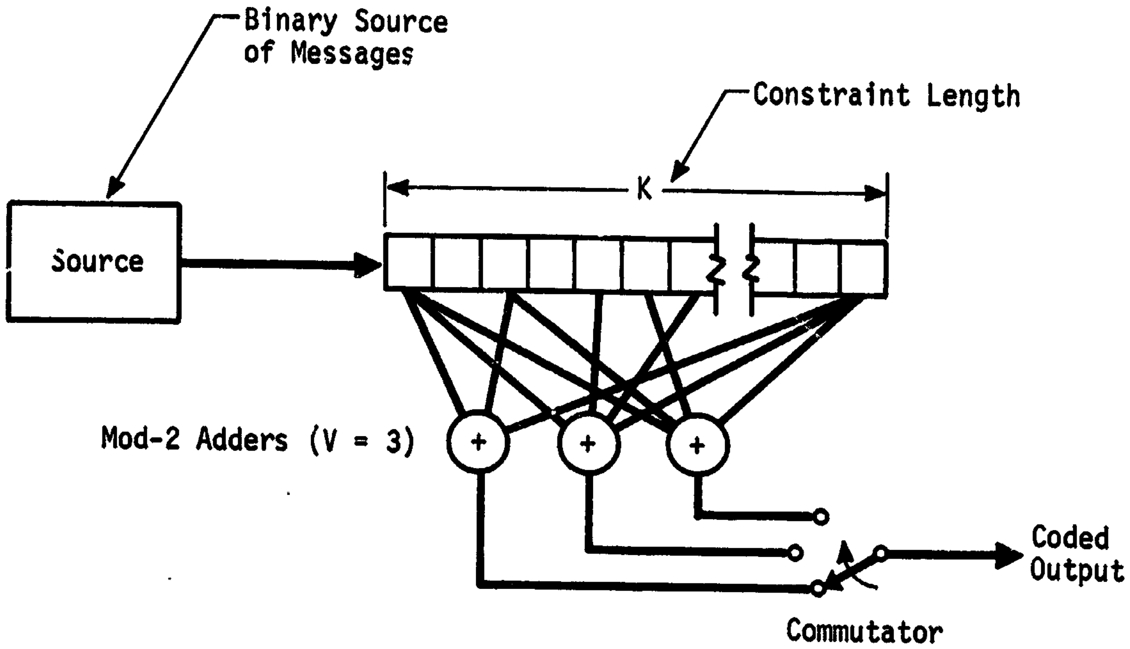


Fig. VI-1 Convolutional Encoder

Heller* gives a method for determining an upper bound on the minimum weight of a convolutional code for any constraint length K and rate 1/V. This bound is based on the total weight of all nonzero code words of a given length and is equal to half the length times the number of nonzero words. The minimum weight for the nonzero words cannot be more than the average weight for these words, which is therefore an upper bound on the minimum weight. This bound is valid for any message length L and is given by

$$\overline{W} = \frac{(K + L - 1)\ 2^{L-1}\ V}{2^{L} - 1} \qquad \text{[VI-1]}$$

The bound is computed for each L starting with 1 until a minimum is found, and this minimum is then an upper bound on the minimum weight. These bounds are mostly noninteger fractions and can therefore be reduced to the integer parts.

A further reduction in the bound is achieved for some cases when the method just described yields an odd number. The reasoning follows from the fact that if a code has any words of odd weight, exactly half of the words have odd weight. Consider for example V = 3, K = 8 for which the minimum value for Eq [VI-1] is found to be 17 1/7 for L = 3. This is the average weight for the seven nonzero code words corresponding to messages of length L = 3. To achieve this value with a minimum weight of 17 would require six words of weight 17 and one of weight 18, obviously contradicting the requirement that the number of words of odd weight must be 0 or 4. The maximum minimum weight is therefore not greater than 16. A similar result is found for V = 2, K = 6.

With this refinement, our computer found good codes that reached the bound in all cases except V = 2, K = 5 and V = 2, K = 8. In the first case the only code with minimum weight 8 is catastrophic. For V = 2, K = 8, good codes of minimum weight 11 were found when the message length L was limited to 6, but all of these were eliminated when L was extended to 10.

---

*J. A. Heller: "Short Constraint Length Convolutional Codes." JPL Space Programs Summary, Vol III, p 37-54.

Table VI-1 shows the minimum weights that were achieved.

Table VI-1 Minimum Weights for Good Codes

V \ K	4	5	6	7	8	9
2	6	7	8	10	10	12
3	10	12	13	15	16	*
4	12	16	18	20	22	*

*These cases were not searched.

B. CATASTROPHIC CODES

Certain convolutional codes are called "catastrophic" because they produce massive blocks of errors. To understand how this can happen, consider the following example.

For K = 5 and V = 2, minimum weight 8 is obtained only with the code generated by the shift register connections 10111, 11101 (and symmetrical transformations of this code). Unfortunately, it turns out that this code is not a good one. Because there are an even number of connections to each mod-2 adder, the coded bit stream is the same for the complement of the message as it is for the message itself. For instance, there is no way to distinguish between the messages...000000... and ...11111.... Simulation on a general-purpose computer confirms this prediction. When the message is all zeros, the decoder puts out long strings of zeros and ones. The simulated noise flips the decoder from one state to the other.

More general considerations permit a quick test to eliminate degenerate codes like the one just described. It is convenient to treat the connection vectors as "polynominals with coefficients from the field of two elements." These "elements" are, of course, 0 and 1 and they obey the rules $0 + 0 = 0$; $0 + 1 = 1$; $1 + 1 = 0$; $0 \cdot 0 = 0$; $0 \cdot 1 = 0$; $1 \cdot 1 = 1$. In polynominal form, the connection vectors for the code discussed above are:

$$x^4 + x^2 + x + 1, \text{ and}$$

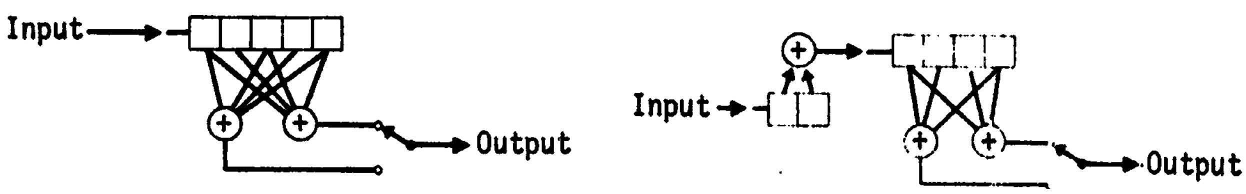
$$x^4 + x^3 + x^2 + 1.$$

These polynominals can be factored as:

$$(x^4 + x^2 + x + 1) = (x + 1)(x^3 + x^2 + 1), \quad [VI-2]$$

$$(x^4 + x^3 + x^2 + 1) = (x + 1)\ (x^3 + x + 1). \quad [VI-3]$$

The fact that both polynominals have x + 1 as a factor indicates that the code is degenerate and is equivalent to a K = 4 code. This can be verified by putting messages through the two encoders in Fig. VI-2. The two-stage shift register that precedes the K = 4 encoder performs a transformation in the message space without affecting the performance of the code.



(a) K = 5 Encoder (b) K = 4 Equivalent

Fig. VI-2 A Degenerate K = 5 Encoder and its K = 4 Equivalent

In general, a code is catastrophic if and only if all of its connection polynominals have a common factor. An important function of the search program is to make this test.

C. THE UNION BOUND

When two or more codes for a given V and K have the same minimum weight, a sharper method of code selection is needed. The simplest improvement is to choose the code with the smallest number of words of minimum weight, and if there is a tie, words of the next higher weight can be counted until a single "best" code is found.

A better way is to make a quantitative choice by using the union bound. This is an upper bound on error probability for maximum likelihood decoding. The probability of each possible error pattern is computed as though the decoder were forced to choose between the correct message and a message with that particular error pattern. All the individual probabilities are then added to give the union bound. At low signal-to-noise ratio the error patterns overlap and the true probability is significantly below the bound, but when the error probability is low, it is a good approximation.

For a given error pattern m starting on a given message bit, the probability is

$$P_m = \frac{1}{\sqrt{2\pi}} \int_{Y_m}^{\infty} \exp(-x^2/2)\, dx. \qquad \text{[VI-4]}$$

Y_m is determined by the signal-to-noise ratio per bit E_b/N_o, the rate denominator V, and the weight W_m of the code word corresponding to the error pattern:

$$Y_m^2 = 2W_m E_b/VN_o. \qquad \text{[VI-5]}$$

The bound on the bit error frequency is found by multiplying each P_m by the number N_m of bit errors in the pattern and summing.

$$P(\text{bit}) < \sum N_m P_m. \qquad \text{[VI-6]}$$

The bound on burst error frequency is also computed. For this calculation we have defined a burst as a pattern of errors that spans no more than 10 bits. The 10-bit limit was chosen to limit the amount of machine computation that was needed. The limit for bursts is similar to Eq [VI-6] but without the multipliers N_m:

$$P(\text{burst}) < \sum P_m. \qquad \text{[VI-7]}$$

The number of codes to be examined increases exponentially with both K and V. For a given K there are K - 2 connections to be chosen for each connection vector if it is assumed that the first and last connections are always used. Thus there are $2^{(K-2)V}/V!$ possible codes that must be tried if permutations of connection vectors are not searched separately. At V = 3 and K = 8, for instance, about 25,000 codes must be processed.

To keep computer time reasonable, the length of the message was limited to 10 bits and code words of weight greater than 2V in excess of the minimum weight were disregarded. For this reason the computed union bound is slightly lower than the correct value. An estimate of the size of the defect may be made by computing the bound using only the shorter messages and lower weights. For instance for the V = 3 K = 8 code 143,185,237 there are two words of weight 16 and eight of weight 18 that were found by the computer in its search. The two at 16 are found when the message length is 2 bits or under and seven of the eight at 18 are found with messages of 5 bits or fewer. It is unlikely that messages longer than 10 bits will yield any more code words of these weights. At 4 dB the two words at 16 contribute 48% of the computed burst-error bound and the eight words at 18 contribute an additional 35%, leaving 17% for words at a weight of 22 (this code has no words of odd weight). If contributions at the higher weights continue to decrease by a factor of 2 for each increase in weight, the computed bound is about 20% low. This represents an error of less the 0.1 dB in signal-to-noise ratio. At 6 dB the error is very much less. The relative performance of the codes is probably indicated quite closely by the bounds as computed.

Figure VI-3 compares simulation results with the union bound. The discontinuity between the two curves is largely explained by the loss due to quantizing to 8 levels for the simulation.

D. DESCRIPTION OF PROGRAM

The strategy of the program is to eliminate bad codes early with a minimum of work. The simplest test is to count the number of shift register connections and compare it to the putative maximum minimum weight. (The number of connections is equal to the weight of the code word corresponding to the message ...0001000...) Next, a check is made to see whether the reverse of the code, which has the same weight structure, was examined earlier.

Thirdly, the code is checked to determine that its weight is equal to or greater than the maximum minimum weight for message vectors out to 111111. Next, by a process of polynomial division, it is found that there are no common factors in the code connection vectors, and then the weights are checked out to message vectors of 1111111111.

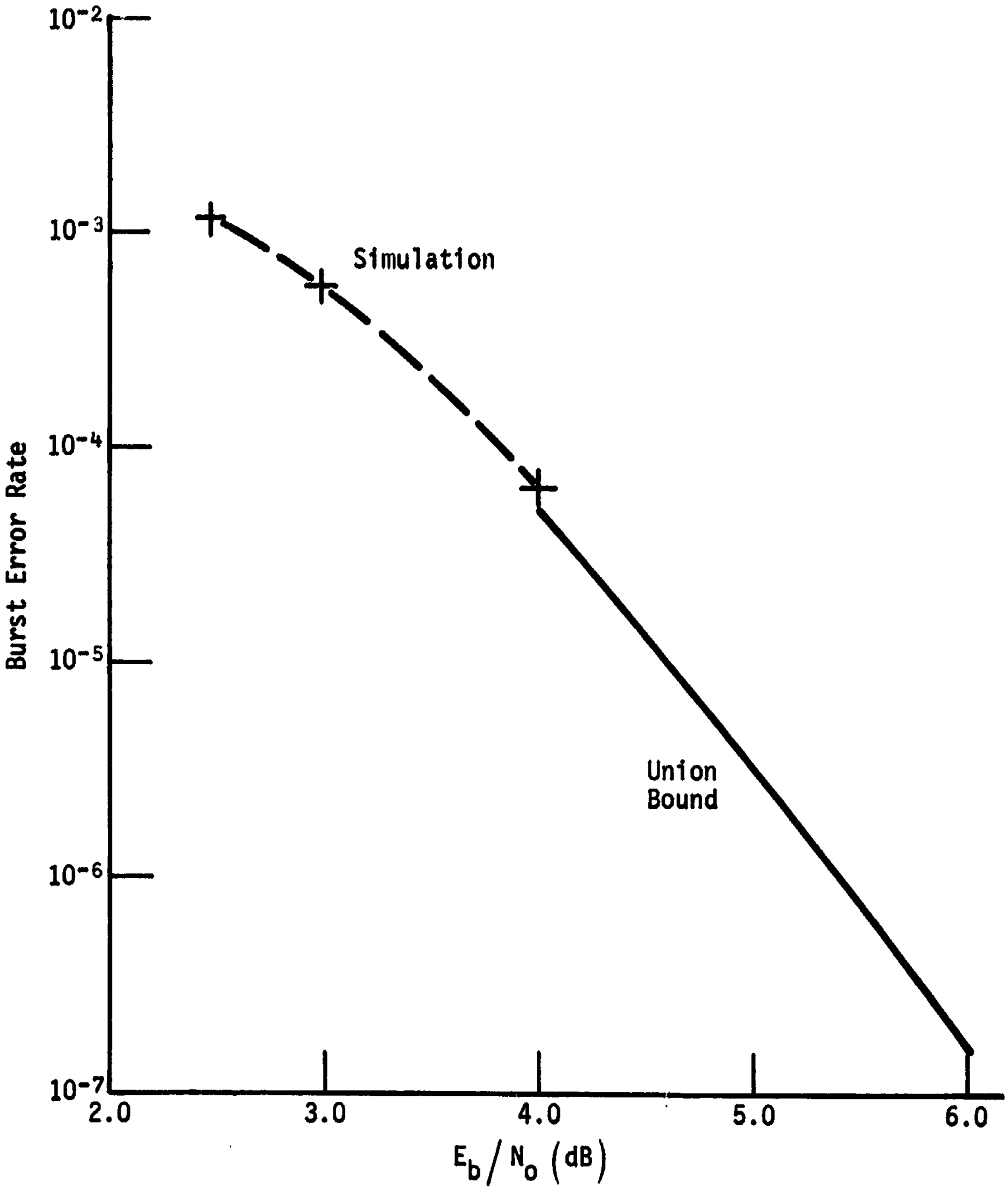


Fig. VI-3 Performance of V = 2, K = 5 Code 10011, 11101

The first assembler program, which searches out the good codes and writes them on tape, is followed by a Fortran program that recognizes the first word in the first record as the rate denominator to be worked with. The Fortran program then computes the approximation

$$\frac{1}{\sqrt{2\pi}} \int_{Y}^{\infty} \exp(-x^2/2)\, dx \approx \frac{(Y^{-1} - Y^{-3} + 3Y^{-5})}{\sqrt{2\pi}} \exp(-Y^2/2) \qquad \text{[VI-8]}$$

where

$$Y = \sqrt{2W_m\, E_s/N_o} \text{ and } E_s = E_b/V$$

and sets up a table of all solutions for the specified V and range of weight values. These are not only stored for the program but are written on the output tape as floating point numbers. The Fortran program transfers the input records it does not process intact to the output tape.

The program then processes all good code records, multiplying the weights in the source record by the proper entry in the error integral solutions table, and solving for the union bound for all values of E_b/N_o from 4.0 dB thru 6.0 dB in increments of 0.2 dB (21 values) for both bit and burst errors. The program outputs four records for each input record:

1) Source data and all error integral data for bit errors;
2) All 21 values of the union bound for bit errors, formatted for printer;
3) Source data and all error integral solutions for burst errors;
4) All 21 values of the union bound for burst errors.

The output tape now has not only source data, but Fortran-processed, expanded data for good codes.

The final program module in the series is an assembler program capable of giving a printer output, sorting, culling, and consolidating all data and giving up to two tape outputs. As used, only one tape was written and it is essentially the same as the Fortran processed tape except that the Fortran program works an entire rate denominator value as a file while the final program puts each K into a file for easier and faster access to smaller areas of data.

E. RESULTS

The top block of information on Table VI-2 of the following printouts represent the consolidated results for each K of each V processed. In order on the page are the best (up to 10) codes surveyed for each V and K, with the connection vectors (in decimal notation) and the value of the union bound at 4 dB on the left; to the right are the same data at the 6 dB point, for bit error computations. An identical selection and presentation for burst error data is given in the center block of each page of Table VI-2.

The lower block on each page of Table VI-2 shows the union bound approximations for not only the best, but also the worst codes that met the selection criteria. The first two pair are the best and worst codes for bit errors at 4 dB and the two remaining pairs are for burst errors. In each case the union bound is tabulated from 4 to 6 dB in increments of 0.2 dB.

Because of the large number of good codes that were found with certain combinations of V and K, the search was stopped before all possibilities were exhausted. The cases that are incomplete are for V = 3, K = 8, which was stopped after 820 codes were found, and V = 4, K = 8, which was stopped after 270. Thus the "best" and "worst" codes are the best and worst of those tried.

Table VI-3 is an incomplete list of the good codes. Each code appears on two lines. The first line shows the aggregate weights of all the code words having minimum weight, minimum weight plus 1, and minimum weight plus 2. Following this are the values of the union bound for bit errors at 4 and 6 dB. The second line for each code shows the number of words at each of the three weights and the union bound for burst errors.

Table VI-2 Consolidated Results

BEST TEN UNION BOUNDS AND A CODE HAVING THESE BOUNDS FOR....
V= 2 K= 4 MESSAGE VECTORS=1-1111111111 MAXIMUM-MINIMUM WEIGHT= 6

4 AND 6 DB TABLES FOR BIT ERRORS

11	15	0.3505E-03	11	15	0.1699E-05
11	13	0.4072E-03	11	13	0.2393E-05

4 AND 6 DB TABLES FOR BURST ERRORS

11	15	0.1294E-03	11	15	0.7634E-06
11	13	0.1521E-03	11	13	0.1112E-05

BEST AND WORST CODES-4 AND 6 DB-BIT AND BURST ERRORS-IN ORDER

11	15		0.3505E-03	0.2211E-03	0.1375E-03	0.8424E-04	0.5081E-04
		0.3016E-04	0.1760E-04	0.1010E-04	0.5685E-05	0.3139E-05	0.1699E-05
11	13		0.4072E-03	0.2602E-03	0.1641E-03	0.1022E-03	0.6272E-04
		0.3795E-04	0.2261E-04	0.1326E-04	0.7636E-05	0.4317E-05	0.2393E-05
11	15		0.3505E-03	0.2211E-03	0.1375E-03	0.8424E-04	0.5081E-04
		0.3016E-04	0.1760E-04	0.1010E-04	0.5685E-05	0.3139E-05	0.1699E-05
11	13		0.4072E-03	0.2602E-03	0.1641E-03	0.1022E-03	0.6272E-04
		0.3795E-04	0.2261E-04	0.1326E-04	0.7636E-05	0.4317E-05	0.2393E-05
11	15		0.1294E-03	0.8357E-04	0.5317E-04	0.3331E-04	0.2053E-04
		0.1244E-04	0.7405E-05	0.4326E-05	0.2478E-05	0.1390E-05	0.7634E-06
11	13		0.1521E-03	0.9991E-04	0.6476E-04	0.4138E-04	0.2604E-04
		0.1613E-04	0.9818E-05	0.5870E-05	0.3442E-05	0.1978E-05	0.1112E-05
11	15		0.1294E-03	0.8357E-04	0.5317E-04	0.3331E-04	0.2053E-04
		0.1244E-04	0.7405E-05	0.4326E-05	0.2478E-05	0.1390E-05	0.7634E-06
11	13		0.1521E-03	0.9991E-04	0.6476E-04	0.4138E-04	0.2604E-04
		0.1613E-04	0.9818E-05	0.5870E-05	0.3442E-05	0.1978E-05	0.1112E-05

Table VI-2 (cont)

BEST TEN UNION BOUNDS AND A CODE HAVING THESE BOUNDS FOR....

V= 2 K= 5 MESSAGE VECTORS=1-1111111111 MAXIMUM-MINIMUM WEIGHT= 7

4 AND 6 DB TABLES FOR BIT ERRORS

19	29	0.1455E-03	19	29	0.3929E-06
19	27	0.1543E-03	19	27	0.4002E-06
19	23	0.1783E-03	19	23	0.5661E-06

4 AND 6 DB TABLES FOR BURST ERRORS

19	29	0.4834E-04	19	29	0.1618E-06
19	27	0.5411E-04	19	27	0.1723E-06
19	23	0.6257E-04	19	23	0.2264E-06

BEST AND WORST CODES-4 AND 6 DB-BIT AND BURST ERRORS-IN ORDER

19	29		0.1455E-03	0.8802E-04	0.5232E-04	0.3054E-04	0.1749E-04
		0.9821E-05	0.5401E-05	0.2907E-05	0.1529E-05	0.7849E-06	0.3929E-06
19	23		0.1783E-03	0.1093E-03	0.6593E-04	0.3908E-04	0.2275E-04
		0.1299E-04	0.7267E-05	0.3979E-05	0.2129E-05	0.1112E-05	0.5661E-06
19	29		0.1455E-03	0.8802E-04	0.5232E-04	0.3054E-04	0.1749E-04
		0.9821E-05	0.5401E-05	0.2907E-05	0.1529E-05	0.7849E-06	0.3929E-06
19	23		0.1783E-03	0.1093E-03	0.6593E-04	0.3908E-04	0.2275E-04
		0.1299E-04	0.7267E-05	0.3979E-05	0.2129E-05	0.1112E-05	0.5661E-06
19	29		0.4834E-04	0.2991E-04	0.1819E-04	0.1086E-04	0.6364E-05
		0.3654E-05	0.2054E-05	0.1129E-05	0.6060E-06	0.3173E-06	0.1618E-06
19	23		0.6257E-04	0.3902E-04	0.2393E-04	0.1440E-04	0.8508E-05
		0.4924E-05	0.2790E-05	0.1545E-05	0.8358E-06	0.4408E-06	0.2264E-06
19	29		0.4834E-04	0.2991E-04	0.1819E-04	0.1086E-04	0.6364E-05
		0.3654E-05	0.2054E-05	0.1129E-05	0.6060E-06	0.3173E-06	0.1618E-06
19	23		0.6257E-04	0.3902E-04	0.2393E-04	0.1440E-04	0.8508E-05
		0.4924E-05	0.2790E-05	0.1545E-05	0.8358E-06	0.4408E-06	0.2264E-06

Table VI-2 (cont)

BEST TEN UNION BOUNDS AND A CODE HAVING THESE BOUNDS FOR....
V= 2 K= 6 MESSAGE VECTORS=1-1111111111 MAXIMUM-MINIMUM WEIGHT= 8

4 AND 6 DB TABLES FOR BIT ERRORS

37	63	0.4145E-04	37	63	0.5137E-07
35	55	0.4164E-04	37	59	0.5347E-07
37	59	0.4415E-04	35	55	0.5795E-07
35	61	0.4416E-04	35	61	0.5932E-07
47	61	0.4481E-04	37	61	0.6008E-07
37	61	0.4595E-04	43	61	0.6105E-07
37	55	0.5418E-04	47	61	0.7272E-07
43	61	0.5761E-04	37	55	0.7804E-07
47	51	0.6281E-04	47	51	0.8091E-07
45	47	0.6526E-04	45	47	0.8268E-07

4 AND 6 DB TABLES FOR BURST ERRORS

37	59	0.1298E-04	43	61	0.1817E-07
37	63	0.1382E-04	37	59	0.1928E-07
43	61	0.1473E-04	37	63	0.2127E-07
47	61	0.1498E-04	47	51	0.2440E-07
35	55	0.1539E-04	45	47	0.2593E-07
35	61	0.1566E-04	47	61	0.2666E-07
37	61	0.1624E-04	35	55	0.2682E-07
37	55	0.1663E-04	35	61	0.2696E-07
47	51	0.1668E-04	37	61	0.2724E-07
45	47	0.1857E-04	37	55	0.2761E-07

BEST AND WORST CODES-4 AND 6 DB-BIT AND BURST ERRORS-IN ORDER

37	63		0.4145E-04	0.2354E-04	0.1310E-04	0.7134E-05	0.3800E-05
		0.1978E-05	0.1005E-05	0.4979E-06	0.2402E-06	0.1127E-06	0.5137E-07
35	59		0.9227E-04	0.5389E-04	0.3086E-04	0.1732E-04	0.9512E-05
		0.5107E-05	0.2677E-05	0.1368E-05	0.6811E-06	0.3295E-06	0.1547E-06
37	63		0.4145E-04	0.2354E-04	0.1310E-04	0.7134E-05	0.3800E-05
		0.1978E-05	0.1005E-05	0.4979E-06	0.2402E-06	0.1127E-06	0.5137E-07
35	59		0.9227E-04	0.5389E-04	0.3086E-04	0.1732E-04	0.9512E-05
		0.5107E-05	0.2677E-05	0.1368E-05	0.6811E-06	0.3295E-06	0.1547E-06
37	59		0.1298E-04	0.7484E-05	0.4232E-05	0.2344E-05	0.1271E-05
		0.6735E-06	0.3487E-06	0.1761E-06	0.8663E-07	0.4146E-07	0.1928E-07
35	59		0.2414E-04	0.1426E-04	0.8258E-05	0.4683E-05	0.2598E-05
		0.1408E-05	0.7444E-06	0.3835E-06	0.1922E-06	0.9358E-07	0.4419E-07
43	61		0.1473E-04	0.8440E-05	0.4729E-05	0.2589E-05	0.1383E-05
		0.7208E-06	0.3657E-06	0.1805E-06	0.8657E-07	0.4029E-07	0.1817E-07
35	59		0.2414E-04	0.1426E-04	0.8258E-05	0.4683E-05	0.2598E-05
		0.1408E-05	0.7444E-06	0.3835E-06	0.1922E-06	0.9358E-07	0.4419E-07

Table VI-2 (cont)

BEST TEN UNION BOUNDS AND A CODE HAVING THESE BOUNDS FOR....
= 2   K= 7   MESSAGE VECTORS=1-1111111111   MAXIMUM-MINIMUM WEIGHT=10

4 AND 6 DB TABLES FOR BIT ERRORS

79	109	0.1274E-04	79	109	0.5343E-08
93	103	0.1615E-04	93	103	0.6839E-08

4 AND 6 DB TABLES FOR BURST ERRORS

79	109	0.3578E-05	79	109	0.1602E-08
93	103	0.3962E-05	93	103	0.1758E-08

BEST AND WORST CODES-4 AND 6 DB-BIT AND BURST ERRORS-IN ORDER

79	109		0.1274E-04	0.6681E-05	0.3416E-05	0.1701E-05	0.8231E-06
		0.3867E-06	0.1761E-06	0.7758E-07	0.3301E-07	0.1354E-07	0.5343E-08
93	103		0.1615E-04	0.8492E-05	0.4351E-05	0.2170E-05	0.1051E-05
		0.4944E-06	0.2253E-06	0.9928E-07	0.4225E-07	0.1733E-07	0.6839E-08
79	109		0.1274E-04	0.6681E-05	0.3416E-05	0.1701E-05	0.8231E-06
		0.3867E-06	0.1761E-06	0.7758E-07	0.3301E-07	0.1354E-07	0.5343E-08
93	103		0.1615E-04	0.8492E-05	0.4351E-05	0.2170E-05	0.1051E-05
		0.4944E-06	0.2253E-06	0.9928E-07	0.4225E-07	0.1733E-07	0.6839E-08
79	109		0.3578E-05	0.1894E-05	0.9772E-06	0.4904E-06	0.2391E-06
		0.1131E-06	0.5183E-07	0.2296E-07	0.9817E-08	0.4044E-08	0.1602E-08
93	103		0.3962E-05	0.2097E-05	0.1081E-05	0.5421E-06	0.2641E-06
		0.1248E-06	0.5711E-07	0.2527E-07	0.1079E-07	0.4442E-08	0.1758E-08
79	109		0.3578E-05	0.1894E-05	0.9772E-06	0.4904E-06	0.2391E-06
		0.1131E-06	0.5183E-07	0.2296E-07	0.9817E-08	0.4044E-08	0.1602E-08
93	103		0.3962E-05	0.2097E-05	0.1081E-05	0.5421E-06	0.2641E-06
		0.1248E-06	0.5711E-07	0.2527E-07	0.1079E-07	0.4442E-08	0.1758E-08

Table VI-2 (cont)

BEST TEN UNION BOUNDS AND A CODE HAVING THESE BOUNDS FOR....
V= 2 K= 8 MESSAGE VECTORS=1-1111111111 MAXIMUM-MINIMUM WEIGHT=10

4 AND 6 DB TABLES FOR BIT ERRORS

149	251	0.3259E-05	149	251	0.7105E-09
137	223	0.3392E-05	159	229	0.8113E-09
143	221	0.3521E-05	147	247	0.8837E-09
173	243	0.3584E-05	173	243	0.9019E-09
183	253	0.3598E-05	183	253	0.9342E-09
159	229	0.3643E-05	157	203	0.9872E-09
143	229	0.3794E-05	147	253	0.1055E-08
143	205	0.3794E-05	143	229	0.1061E-08
155	245	0.3870E-05	143	221	0.1114E-08
147	253	0.3931E-05	143	205	0.1172E-08

4 AND 6 DB TABLES FOR BURST ERRORS

149	251	0.9247E-06	149	251	0.2464E-09
155	245	0.9286E-06	173	243	0.2464E-09
173	243	0.9314E-06	155	245	0.2564E-09
159	229	0.1029E-05	159	229	0.2779E-09
189	203	0.1036E-05	189	203	0.2805E-09
183	253	0.1086E-05	147	247	0.2978E-09
143	221	0.1090E-05	173	231	0.3374E-09
147	247	0.1112E-05	157	203	0.3439E-09
157	203	0.1128E-05	183	253	0.3636E-09
149	247	0.1150E-05	143	221	0.3736E-09

BEST AND WORST CODES-4 AND 6 DB-BIT AND BURST ERRORS-IN ORDER

149	251		0.3259E-05	0.1612E-05	0.7755E-06	0.3624E-06	0.1644E-06
		0.7226E-07	0.3075E-07	0.1266E-07	0.5029E-08	0.1927E-08	0.7105E-09
203	251		0.1537E-04	0.8101E-05	0.4162E-05	0.2080E-05	0.1010E-05
		0.4762E-06	0.2174E-06	0.9599E-07	0.4092E-07	0.1681E-07	0.6643E-08
149	251		0.3259E-05	0.1612E-05	0.7755E-06	0.3624E-06	0.1644E-06
		0.7226E-07	0.3075E-07	0.1266E-07	0.5029E-08	0.1927E-08	0.7105E-09
203	251		0.1537E-04	0.8101E-05	0.4162E-05	0.2080E-05	0.1010E-05
		0.4762E-06	0.2174E-06	0.9599E-07	0.4092E-07	0.1681E-07	0.6643E-08
149	251		0.9247E-06	0.4650E-06	0.2276E-06	0.1083E-06	0.5007E-07
		0.2245E-07	0.9756E-08	0.4101E-08	0.1666E-08	0.6529E-09	0.2464E-09
155	251		0.3182E-05	0.1689E-05	0.8733E-06	0.4393E-06	0.2146E-06
		0.1017E-06	0.4667E-07	0.2070E-07	0.8863E-08	0.3655E-08	0.1449E-08
149	251		0.9247E-06	0.4650E-06	0.2276E-06	0.1083E-06	0.5007E-07
		0.2245E-07	0.9756E-08	0.4101E-08	0.1666E-08	0.6529E-09	0.2464E-09
155	251		0.3182E-05	0.1689E-05	0.8733E-06	0.4393E-06	0.2146E-06
		0.1017E-06	0.4667E-07	0.2070E-07	0.8863E-08	0.3655E-08	0.1449E-08

Table VI-2 (cont)

BEST TEN UNION BOUNDS AND A CODE HAVING THESE BOUNDS FOR....
V= 2 K= 9 MESSAGE VECTORS=1-111111111 MAXIMUM-MINIMUM WEIGHT=12

4 AND 6 DB TABLES FOR BIT ERRORS

285	431	0.9773E-06	285	431	0.8667E-10
285	375	0.1035E-05	285	375	0.9360E-10
307	493	0.1035E-05	307	493	0.9728E-10
299	493	0.1096E-05	299	493	0.1027E-09
285	367	0.1128E-05	285	367	0.1072E-09
313	443	0.1133E-05	303	493	0.1076E-09
303	493	0.1136E-05	301	487	0.1079E-09
309	505	0.1142E-05	309	505	0.1094E-09
301	487	0.1152E-05	333	447	0.1099E-09
309	491	0.1186E-05	313	443	0.1110E-09

4 AND 6 DB TABLES FOR BURST ERRORS

307	493	0.2757E-06	333	447	0.2592E-10
333	447	0.2787E-06	303	493	0.2725E-10
299	493	0.2813E-06	307	493	0.2763E-10
285	431	0.2824E-06	299	493	0.2779E-10
303	493	0.2824E-06	285	431	0.2783E-10
301	487	0.2840E-06	301	487	0.2787E-10
301	499	0.2986E-06	301	499	0.3010E-10
285	375	0.3008E-06	285	375	0.3015E-10
309	505	0.3010E-06	309	505	0.3015E-10
285	367	0.3020E-06	285	367	0.3019E-10

BEST AND WORST CODES-4 AND 6 DB-BIT AND BURST ERRORS-IN ORDER

285	431		0.9773E-06	0.4510E-06	0.2018E-06	0.8737E-07	0.3657E-07
		0.1477E-07	0.5744E-08	0.2147E-08	0.7701E-09	0.2644E-09	0.8667E-10
333	375		0.2010E-05	0.9495E-06	0.4343E-06	0.1921E-06	0.8202E-07
		0.3374E-07	0.1335E-07	0.5070E-08	0.1844E-08	0.6413E-09	0.2127E-09
285	431		0.9773E-06	0.4510E-06	0.2018E-06	0.8737E-07	0.3657E-07
		0.1477E-07	0.5744E-08	0.2147E-08	0.7701E-09	0.2644E-09	0.8667E-10
333	375		0.2010E-05	0.9495E-06	0.4343E-06	0.1921E-06	0.8202E-07
		0.3374E-07	0.1335E-07	0.5070E-08	0.1844E-08	0.6413E-09	0.2127E-09
307	493		0.2757E-06	0.1292E-06	0.5866E-07	0.2577E-07	0.1094E-07
		0.4473E-08	0.1761E-08	0.6657E-09	0.2412E-09	0.8357E-10	0.2763E-10
333	375		0.4380E-06	0.2072E-06	0.9493E-07	0.4205E-07	0.1797E-07
		0.7403E-08	0.2932E-08	0.1114E-08	0.4058E-09	0.1412E-09	0.4684E-10
333	447		0.2787E-06	0.1300E-06	0.5869E-07	0.2562E-07	0.1080E-07
		0.4384E-08	0.1712E-08	0.6418E-09	0.2305E-09	0.7914E-10	0.2592E-10
333	375		0.4380E-06	0.2072E-06	0.9493E-07	0.4205E-07	0.1797E-07
		0.7403E-08	0.2932E-08	0.1114E-08	0.4058E-09	0.1412E-09	0.4684E-10

Table VI-2 (cont)

BEST TEN UNION BOUNDS AND A CODE HAVING THESE BOUNDS FOR....
V= 3 K= 4 MESSAGE VECTORS=1-1111111111 MAXIMUM-MINIMUM WEIGHT=10

4 AND 6 DB TABLES FOR BIT ERRORS

11	13	15	0.2013E-03	11	13	15	0.8606E-06

4 AND 6 DB TABLES FOR BURST ERRORS

11	13	15	0.8400E-04	11	13	15	0.4131E-06

BEST AND WORST CODES-4 AND 6 DB-BIT AND BURST ERRORS-IN ORDER

11	13	15	0.2013E-03	0.1263E-03	0.7801E-04	0.4744E-04	0.2838E-04
		0.1668E-04	0.9618E-05	0.5438E-05	0.3010E-05	0.1629E-05	0.8606E-06
11	13	15	0.2013E-03	0.1263E-03	0.7801E-04	0.4744E-04	0.2838E-04
		0.1668E-04	0.9618E-05	0.5438E-05	0.3010E-05	0.1629E-05	0.8606E-06
11	13	15	0.2013E-03	0.1263E-03	0.7801E-04	0.4744E-04	0.2838E-04
		0.1668E-04	0.9618E-05	0.5438E-05	0.3010E-05	0.1629E-05	0.8606E-06
11	13	15	0.2013E-03	0.1263E-03	0.7801E-04	0.4744E-04	0.2838E-04
		0.1668E-04	0.9618E-05	0.5438E-05	0.3010E-05	0.1629E-05	0.8606E-06
11	13	15	0.8400E-04	0.5374E-04	0.3384E-04	0.2094E-04	0.1273E-04
		0.7593E-05	0.4438E-05	0.2539E-05	0.1420E-05	0.7757E-06	0.4131E-06
11	13	15	0.8400E-04	0.5374E-04	0.3384E-04	0.2094E-04	0.1273E-04
		0.7593E-05	0.4438E-05	0.2539E-05	0.1420E-05	0.7757E-06	0.4131E-06
11	13	15	0.8400E-04	0.5374E-04	0.3384E-04	0.2094E-04	0.1273E-04
		0.7593E-05	0.4438E-05	0.2539E-05	0.1420E-05	0.7757E-06	0.4131E-06
11	13	15	0.8400E-04	0.5374E-04	0.3384E-04	0.2094E-04	0.1273E-04
		0.7593E-05	0.4438E-05	0.2539E-05	0.1420E-05	0.7757E-06	0.4131E-06

Table VI-2 (cont)

BEST TEN UNION BOUNDS AND A CCDE HAVING THESE BOUNDS FOR....
V= 3 K= 5 MESSAGE VECTORS=1-1111111111 MAXIMUM-MINIMUM WEIGHT=12
4 AND 6 DB TABLES FOR BIT ERRORS

21	27	31	0.6444E-04	21	27	31	0.1093E-06

4 AND 6 DB TABLES FOR BURST ERRORS

21	27	31	0.2305E-C4	21	27	31	0.4393E-07

BEST AND WORST CODES-4 AND 6 DB-BIT AND BURST ERRORS-IN ORDER

21 27 31		0.6444E-04	0.3758E-04	0.2151E-04	0.1207E-04	0.6638E-05
	0.3570E-05	0.1875E-05	0.9605E-06	0.4791E-06	0.2323E-06	0.1[illegible]3E-06
21 27 31		0.6444E-04	0.3758E-04	0.2151E-04	0.1207E-04	0.[illegible]638E-05
	0.3570E-05	0.1875E-05	0.9605E-06	0.4791E-06	0.2323E-06	0.1093E-06
21 27 31		0.6444E-04	0.3758E-04	0.2151E-C4	0.1207E-04	0.6638E-05
	0.3570E-05	0.1875E-05	0.9605E-06	0.4791E-06	0.2323E-06	0.1093E-06
21 27 31		0.6444E-04	0.3758E-04	0.2151E-04	0.1207E-04	0.6638E-05
	0.3570E-05	0.1875E-05	0.9605E-06	C.4791E-06	0.2323E-06	0.1093E-06
21 27 31		0.2305E-04	0.1368E-04	0.7957E-05	0.45[illegible]3E-05	0.2526E-05
	0.1375E-05	0.7300E-06	0.3776E-06	0.1899E-06	0.9276E-C7	0.4393E-07
21 27 31		0.2305E-04	0.1368E-04	0.7957E-05	0.4533E-05	0.2526E-05
	0.1375E-05	0.7300E-06	0.3776E-06	0.1899E-06	0.9276E-07	0.4393E-07
21 27 31		0.2305E-04	C.1368E-04	0.7957E-05	0.4533E-05	0.2526E-05
	0.1375E-C5	0.7300E-06	0.3776E-06	0.1899E-06	0.9276E-07	0.4393E-07
21 27 31		0.2305E-04	0.1368E-04	0.7957E-05	0.4533E-C5	0.2526E-05
	0.1375E-05	0.7300E-06	0.3776E-C6	0.1899E-06	0.9276E-07	0.4393E-07

Ta e VI-2 (cont)

BEST TEN UNION BOUNDS AND A CODE HAVING THESE BOUNDS FOR....
V= 3 K= 6 MESSAGE VECTORS=1-1111111111 MAXIMUM-MINIMUM WEIGHT=13

4 AND 6 DB TABLES FOR BIT ERRORS

39	43	61	0.1778E-04	39	43	61	0.1093E-07
39	45	55	0.1832E-04	39	45	55	0.1320E-07
37	47	55	0.1932E-04	37	47	55	0.1552E-07
37	55	59	0.2044E-04	37	55	59	0.1583E-07
39	47	53	0.2111E-04	39	47	53	0.1654E-07
45	47	51	0.2148E-04	45	47	51	0.1669E-07
37	55	61	0.2260E-04	37	55	61	0.1742E-07
37	59	61	0.2267E-04	37	59	61	0.1743E-07
37	47	59	0.2313E-04	39	45	61	0.1806E-07
39	45	61	0.2336E-04	37	47	59	0.1873E-07

4 AND 6 DB TABLES FOR BURST ERRORS

39	43	61	0.6008E-05	39	43	61	0.4794E-08
39	45	55	0.6502E-05	39	45	55	0.5776E-08
37	55	59	0.6711E-05	37	55	59	0.6267E-08
37	47	55	0.7037E-05	45	47	51	0.6608E-08
45	47	51	0.7[illegible]31E-05	37	47	55	0.6623E-08
37	55	61	0.7234E-05	37	55	61	0.6695E-08
39	47	53	0.7387E-05	39	47	53	0.6818E-08
37	47	59	0.7434E-05	37	59	61	0.6890E-08
37	59	61	0.7486E-05	37	47	59	0.7037E-08
39	45	61	0.7867E-05	39	45	61	0.7185E-08

BEST AND WORST CODES-4 AND 6 DB-BIT AND BURST ERRORS-IN ORDER

39	43	61	0.1778E-04	0.9623E-05	0.5080E-05	0.2614E-05	0.1309E-05
		0.6373E-06	0.3013E-06	0.1382E-06	0.6136E-07	0.2635E-07	0.1093E-07
39	43	55	0.2816E-04	0.1560E-04	0.8450E-05	0.4468E-05	0.2304E-05
		0.1158E-05	0.5661E-06	0.2690E-06	0.1240E-06	0.5544E-07	0.2398E-07
39	43	61	0.1778E-04	0.9623E-05	0.5080E-05	0.2614E-05	0.1309E-05
		0.6373E-06	0.3013E-06	0.1382E-06	0.6136E-07	0.2635E-07	0.1093E-07
39	45	47	0.2786E-04	0.1547E-04	0.8392E-05	0.4448E-05	0.2300E-05
		0.1159E-05	0.5685E-06	0.2710E-06	0.1253E-06	0.5617E-07	0.2435E-07
39	43	61	0.6008E-05	0.3324E-05	0.1796E-05	0.9462E-06	0.4857E-06
		0.2426E-06	0.1178E-06	0.5552E-07	0.2537E-07	0.1122E-07	0.4794E-08
39	45	47	0.9425E-05	0.5308E-05	0.2922E-05	0.1570E-05	0.8227E-06
		0.4198E-06	0.2084E-06	0.1004E-06	0.4696E-07	0.2126E-07	0.9307E-08
39	43	61	0.6008E-05	0.3324E-05	0.1796E-05	0.9462E-06	0.4857E-06
		0.2426E-06	0.1178E-06	0.5552E-07	0.2537E-07	0.1122E-07	0.4794E-08
39	45	47	0.9425E-05	0.5308E-05	0.2922E-05	0.1570E-05	0.8227E-06
		0.4198E-06	0.2084E-06	0.1004E-06	0.4696E-07	0.2126E-07	0.9307E-08

Table VI-2 (cont)

BEST TEN UNION BOUNDS AND A CODE HAVING THESE BOUNDS FOR....
V= 3 K= 7 MESSAGE VECTORS=1-1111111111 MAXIMUM-MINIMUM WEIGHT=15

4 AND 6 DB TABLES FOR BIT ERRORS

79	87	109	0.5364E-05	79	87	109	0.1607E-08
79	93	109	0.5443E-05	79	93	109	0.1630E-08
83	93	111	0.5600E-05	83	93	111	0.1764E-08
75	93	111	0.5796E-05	83	109	125	0.1863E-08
77	93	111	0.5844E-05	75	93	111	0.2002E-08
83	109	125	0.6055E-05	77	93	111	0.2061E-08
85	103	119	0.6248E-05	85	103	119	0.2151E-08
79	107	117	0.6842E-05	87	103	117	0.2366E-08
77	107	125	0.6934E-05	77	107	125	0.2379E-08
87	103	117	0.6948E-05	79	107	117	0.2397E-08

4 AND 6 DB TABLES FOR BURST ERRORS

83	93	111	0.1655E-05	83	93	111	0.5778E-09
79	87	109	0.1739E-05	79	87	109	0.6108E-09
79	93	109	0.1800E-05	79	93	109	0.6316E-09
79	91	117	0.1862E-05	79	91	117	0.6476E-09
83	109	125	0.1887E-05	83	109	125	0.6695E-09
83	95	109	0.1889E-05	83	95	109	0.6696E-09
75	93	111	0.1954E-05	75	93	111	0.7570E-09
77	93	111	0.2005E-05	79	107	117	0.7723E-09
87	103	117	0.2007E-05	87	103	117	0.7764E-09
79	107	117	0.2036E-05	77	107	125	0.7766E-09

BEST AND WORST CODES-4 AND 6 DB-BIT AND BURST ERRORS-IN ORDER

79	87	109	0.5364E-05	0.2727E-05	0.1350E-05	0.6499E-06	0.3040E-06
		0.1380E-06	0.6070E-07	0.2583E-07	0.1062E-07	0.4210E-08	0.1607E-08
75	117	119	0.8278E-05	0.4289E-05	0.2165E-05	0.1063E-05	0.5071E-06
		0.2347E-06	0.1053E-06	0.4565E-07	0.1912E-07	0.7715E-08	0.2995E-08
79	87	109	0.5364E-05	0.2727E-05	0.1350E-05	0.6499E-06	0.3040E-06
		0.1380E-06	0.6070E-07	0.2583E-07	0.1062E-07	0.4210E-08	0.1607E-08
83	111	117	0.8243E-05	0.4287E-05	0.2172E-05	0.1071E-05	0.5129E-06
		0.2384E-06	0.1074E-06	0.4676E-07	0.1967E-07	0.7974E-08	0.3110E-08
83	93	111	0.1655E-05	0.8545E-06	0.4298E-06	0.2103E-06	0.9993E-07
		0.4608E-07	0.2058E-07	0.8893E-08	0.3711E-08	0.1493E-08	0.5778E-09
75	117	119	0.2366E-05	0.1239E-05	0.6320E-06	0.3135E-06	0.1510E-06
		0.7050E-07	0.3188E-07	0.1393E-07	0.5878E-08	0.2388E-08	0.9331E-09
83	93	111	0.1655E-05	0.8545E-06	0.4298E-06	0.2103E-06	0.9993E-07
		0.4608E-07	0.2058E-07	0.8893E-08	0.3711E-08	0.1493E-08	0.5778E-09
75	117	119	0.2366E-05	0.1239E-05	0.6320E-06	0.3135E-06	0.1510E-06
		0.7050E-07	0.3188E-07	0.1393E-07	0.5878E-08	0.2388E-08	0.9331E-09

Table VI-2 (cont)

BEST TEN UNION BOUNDS AND A CODE HAVING THESE BOUNDS FOR....
V= 3 K= 8 MESSAGE VECTORS=1-1111111111 MAXIMUM-MINIMUM WEIGHT=16

4 AND 6 DB TABLES FOR BIT ERRORS

149	217	247	0.1137E-05	149	217	247	0.1138E-09
143	185	237	0.1233E-05	153	173	239	0.1353E-09
153	173	239	0.1239E-05	149	217	251	0.1802E-09
149	217	251	0.1258E-05	143	185	237	0.1806E-09
153	187	253	0.1301E-05	151	211	253	0.1882E-09
147	185	253	0.1307E-05	155	167	249	0.1926E-09
155	167	249	0.1328E-05	139	191	205	0.1975E-09
139	191	205	0.1347E-05	151	211	249	0.2000E-09
141	191	235	0.1366E-C5	141	179	253	0.2035E-09
139	191	221	0.1368E-05	153	187	253	0.2081E-09

4 AND 6 DB TABLES FOR BURST ERRORS

149	217	247	0.3861E-06	149	217	247	0.5941E-10
153	173	239	0.4093E-06	153	173	239	0.6352E-10
151	211	253	0.4502E-06	151	211	253	0.7582E-10
149	217	251	0.4689E-06	151	181	251	0.8259E-10
147	215	253	0.4751E-06	151	203	253	0.8514E-10
143	185	237	0.4753E-06	147	215	253	0.8604E-10
155	167	249	0.4757E-06	151	221	243	0.8718E-10
149	179	247	0.4769E-06	151	173	223	0.8779E-10
151	173	243	0.4771E-06	149	183	251	0.9013E-10
151	181	251	0.4819E-06	149	215	247	0.9052E-10

BEST AND WORST CODES-4 AND 6 DB-BIT AND BURST ERRORS-IN ORDER

149	217	247	0.1137E-05	0.5270E-06	0.2368E-06	0.1031E-06	0.4342E-07
		0.1768E-07	0.6949E-08	0.2634E-08	0.9616E-C9	0.3376E-09	0.1138E-09
155	157	159	0.5011E-05	0.2516E-05	0.1228E-05	0.5820E-06	0.2675E-06
		0.1190E-06	0.5121E-07	0.2126E-07	0.8500E-08	0.3268E-08	0.1205E-08
149	217	247	0.1137E-05	0.5270E-06	0.2368E-06	0.1031E-06	0.4342E-07
		0.1768E-07	0.6949E-08	0.2634E-08	0.9616E-09	0.3376E-09	0.1138E-09
155	157	159	0.5011E-05	0.2516E-05	0.1228E-05	0.5820E-06	0.2675E-06
		0.1190E-06	0.5121E-07	0.2126E-07	0.8500E-08	0.3268E-08	0.1205E-08
149	217	247	0.3861E-06	0.1852E-06	0.8630E-07	0.3905E-07	0.1713E-07
		0.7280E-08	0.2992E-08	0.1188E-08	0.4548E-09	0.1677E-09	0.5941E-10
155	157	159	0.1458E-05	0.7406E-06	0.3657E-06	0.1753E-06	0.8148E-07
		0.3665E-07	0.1593E-07	0.6676E-08	0.2694E-08	0.1045E-08	0.3887E-09
149	217	247	0.3861E-06	0.1852E-06	0.8630E-07	0.3905E-07	0.1713E-07
		0.7280E-08	0.2992E-08	0.1188E-08	0.4548E-09	0.1677E-09	0.5941E-10
155	157	159	0.1458E-05	0.7406E-06	0.3657E-06	0.1753E-06	0.8148E-07
		0.3665E-07	0.1593E-07	0.6676E-08	0.2694E-08	0.1045E-08	0.3887E-09

Table VI-2 (cont)

BEST TEN UNION BOUNDS AND A CODE HAVING THESE BOUNDS FOR....
V= 4 K= 4 MESSAGE VECTORS=1-1111111111 MAXIMUM-MINIMUM WEIGHT=12

4 AND 6 CB TABLES FOR BIT ERRORS

9	11	13	15	0.1323E-03	9	11	13	15	0.7227E-06

4 AND 6 DB TABLES FOR BURST ERRORS

9	11	13	15	0.8471E-04	9	11	13	15	0.6083E-06

BEST AND WORST CODES-4 AND 6 CB-BIT AND BURST ERRORS-IN ORDER

9	11	13	15	0.1323E-03	0.8434E-04	0.5303E-04	0.3286E-04	0.2006E-04
		0.1205E-04		0.7121E-05	0.4136E-05	0.2358E-05	0.1319E-05	0.7227E-06
9	11	13	15	0.1323E-03	0.8434E-04	0.5303E-04	0.3286E-04	0.2006E-04
		0.1205E-04		0.7121E-05	0.4136E-05	0.2358E-05	0.1319E-05	0.7227E-06
9	11	13	15	0.1323E-03	0.8434E-04	0.5303E-04	0.3286E-04	0.2006E-04
		0.1205E-04		0.7121E-05	0.4136E-05	0.2358E-05	0.1319E-05	0.7227E-06
9	11	13	15	0.1323E-03	0.8434E-04	0.5303E-04	0.3286E-04	0.2006E-04
		0.1205E-04		0.7121E-05	0.4136E-05	0.2358E-05	0.1319E-05	0.7227E-06
9	11	13	15	0.8471E-04	0.5577E-04	0.3618E-04	0.2312E-04	0.1454E-04
		0.8988E-05		0.5459E-05	0.3253E-05	0.1901E-05	0.1087E-05	0.6083E-06
9	11	13	15	0.8471E-04	0.5577E-04	0.3618E-04	0.2312E-04	0.1454E-04
		0.8988E-05		0.5459E-05	0.3253E-05	0.1901E-05	0.1087E-05	0.6083E-06
9	11	13	15	0.8471E-04	0.5577E-04	0.3618E-04	0.2312E-04	0.1454E-04
		0.8988E-05		0.5459E-05	0.3253E-05	0.1901E-05	0.1087E-05	0.6083E-06
9	11	13	15	0.8471E-04	0.5577E-04	0.3618E-04	0.2312E-04	0.1454E-04
		0.8988E-05		0.5459E-05	0.3253E-05	0.1901E-05	0.1087E-05	0.6083E-06

Table VI-2 (cont)

```
BEST TEN UNION BOUNDS AND A CCDE HAVING THESE BOUNDS FOR....
V= 4    K= 5    MESSAGE VECTORS=1-1111111111    MAXIMUM-MINIMUM WEIGHT=16
4 AND 6 CB TABLES FOR BIT ERRORS
      21   23   27   31    0.4828E-04          21   23   27   31    0.7804E-07

4 AND 6 DB TABLES FOR BURST ERRORS
      21   23   27   31    0.1953E-04          21   23   27   31    0.3640E-07

BEST AND WORST CODES-4 AND 6 DB-BIT AND BURST ERRORS-IN ORDER
 21   23   27   31     0.4828E-04  0.2808E-04  0.1602E-04  0.8954E-05  0.4900E-05
           0.2621E-05  0.1369E-05  0.6976E-06  0.3460E-06  0.1668E-06  0.7804E-07
 21   23   27   31     0.4828E-04  0.2808E-04  0.1602E-04  0.8954E-05  0.4900E-05
           0.2621E-05  0.1369E-05  0.6976E-06  0.3460E-06  0.1668E-06  0.7804E-07
 21   23   27   31     0.4828E-04  0.2808E-04  0.1602E-04  0.8954E-05  0.4900E-05
           0.2621E-05  0.1369E-05  0.6976E-06  0.3460E-06  0.1668E-06  0.7804E-07
 21   23   27   31     0.4828E-04  0.2808E-04  0.1602E-04  0.8954E-05  0.4900E-05
           0.2621E-05  0.1369E-05  0.6976E-06  0.3460E-06  0.1668E-06  0.7804E-07
 21   23   27   31     0.1953E-04  0.1158E-04  0.6724E-05  0.3824E-05  0.2126E-05
           0.1155E-05  0.6115E-06  0.3154E-06  0.1582E-06  0.7708E-07  0.3640E-07
 21   23   27   31     0.1953E-04  0.1158E-04  0.6724E-05  0.3824E-05  0.2126E-05
           0.1155E-05  0.6115E-06  0.3154E-06  0.1582E-06  0.7708E-07  0.3640E-07
 21   23   27   31     0.1953E-04  0.1158E-04  0.6724E-05  0.3824E-05  0.2126E-05
           0.1155E-05  0.6115E-06  0.3154E-06  0.1582E-06  0.7708E-07  0.3640E-07
 21   23   27   31     0.1953E-04  0.1158E-04  0.6724E-05  0.3824E-05  0.2126E-05
           0.1155E-05  0.6115E-06  0.3154E-06  0.1582E-06  0.7708E-07  0.3640E-07
```

Table VI-2 (cont)

BEST TEN UNION BOUNDS AND A CODE HAVING THESE BOUNDS FOR....
V= 4 K= 6 MESSAGE VECTORS=1-1111111111 MAXIMUM-MINIMUM WEIGHT=18

4 AND 6 DB TABLES FOR BIT ERRORS

37	45	59	63	0.1180E-04	37	45	59	63	0.7447E-08
37	45	55	63	0.1211E-04	37	45	55	63	0.7763E-08
37	47	55	61	0.1238E-04	37	47	55	61	0.7962E-08
37	43	55	63	0.1315E-04	37	43	55	63	0.8492E-08
37	47	59	61	0.1353E-04	37	47	59	61	0.8561E-08
39	47	53	61	0.1389E-04	37	43	59	63	0.9029E-08
37	43	59	63	0.1413E-04	39	47	53	59	0.9459E-08
39	43	47	61	0.1415E-04	39	43	55	61	0.9657E-08
39	47	53	59	0.1417E-04	39	47	53	61	0.1057E-07
39	43	55	61	0.1488E-04	39	43	47	61	0.1060E-07

4 AND 6 DB TABLES FOR BURST ERRORS

37	45	59	63	0.4848E-05	37	45	59	63	0.3854E-08
37	47	55	61	0.4932E-05	37	47	55	61	0.3951E-08
37	45	55	63	0.5018E-05	37	45	55	63	0.3973E-08
37	43	55	63	0.5047E-05	37	43	55	63	0.4044E-08
39	47	53	59	0.5141E-05	39	47	53	59	0.4076E-08
37	47	59	61	0.5174E-05	37	47	59	61	0.4089E-08
39	43	55	61	0.5253E-05	39	43	55	61	0.4108E-08
37	43	59	63	0.5290E-05	37	43	59	63	0.4178E-08
37	47	55	59	0.5453E-05	37	47	55	59	0.4756E-08
39	47	53	61	0.5587E-05	45	47	51	61	0.4804E-08

BEST AND WORST CODES-4 AND 6 DB-BIT AND BURST ERRORS-IN ORDER

37	45	59 63	0.1180E-04	0.6340E-05	0.3330E-05	0.1708E-05	0.8548E-06
		0.4169E-06	0.1979E-06	0.9128E-07	0.4088E-07	0.1774E-07	0.7447E-08
39	45	47 59	0.1936E-04	0.1068E-04	0.5761E-05	0.3036E-05	0.1560E-05
		0.7815E-06	0.3807E-06	0.1802E-06	0.8269E-07	0.3673E-07	0.1577E-07
37	45	59 63	0.1180E-04	0.6340E-05	0.3330E-05	0.1708E-05	0.8548E-06
		0.4169E-06	0.1979E-06	0.9128E-07	0.4088E-07	0.1774E-07	0.7447E-08
39	45	47 59	0.1936E-04	0.1068E-04	0.5761E-05	0.3036E-05	0.1560E-05
		0.7815E-06	0.3807E-06	0.1802E-06	0.8269E-07	0.3673E-07	0.1577E-07
37	45	59 63	0.4848E-05	0.2675E-05	0.1443E-05	0.7593E-06	0.3896E-06
		0.1946E-06	0.9453E-07	0.4458E-07	0.2038E-07	0.9015E-08	0.3854E-08
39	45	47 59	0.6532E-05	0.3652E-05	0.1995E-05	0.1064E-05	0.5528E-06
		0.2796E-06	0.1375E-06	0.6557E-07	0.3031E-07	0.1355E-07	0.5852E-08
37	45	59 63	0.4848E-05	0.2675E-05	0.1443E-05	0.7593E-06	0.3896E-06
		0.1946E-06	0.9453E-07	0.4458E-07	0.2038E-07	0.9015E-08	0.3854E-08
39	45	47 59	0.6532E-05	0.3652E-05	0.1995E-05	0.1064E-05	0.5528E-06
		0.2796E-06	0.1375E-06	0.6557E-07	0.3031E-07	0.1355E-07	0.5852E-08

Table VI-2 (cont)

BEST TEN UNION BOUNDS AND A CODE HAVING THESE BOUNDS FOR....
V= 4 K= 7 MESSAGE VECTORS=1-11111111 MAXIMUM-MINIMUM WEIGHT=20

4 AND 6 DB TABLES FOR BIT ERRORS

75	95	109	121	0.3134E-05	79	87	109	121	0.8327E-09
79	87	109	121	0.3251E-05	79	95	101	109	0.8553E-09
73	93	107	127	0.3303E-05	79	93	101	123	0.8855E-09
73	93	111	123	0.3358E-05	79	91	101	125	0.9271E-09
79	95	101	109	0.3443E-05	75	95	109	121	0.9440E-09
79	93	101	123	0.3447E-05	75	111	117	121	0.9524E-09
75	87	109	127	0.3513E-05	77	87	121	123	0.9559E-09
79	95	109	117	0.3524E-05	79	93	101	111	0.9779E-09
75	95	109	125	0.3527E-05	75	87	109	127	0.1043E-08
73	87	123	125	0.3592E-05	73	93	111	123	0.1054E-08

4 AND 6 DB TABLES FOR BURST ERRORS

79	87	109	121	0.1240E-05	79	87	109	121	0.4119E-09
79	93	101	123	0.1248E-05	79	93	101	123	0.4135E-09
75	95	109	121	0.1296E-05	79	91	101	125	0.4201E-09
75	87	109	127	0.1298E-05	79	95	101	109	0.4203E-09
79	91	101	125	0.1298E-05	75	111	117	121	0.4305E-09
79	95	101	109	0.1307E-05	77	87	121	123	0.4365E-09
75	111	117	121	0.1319E-05	79	93	101	111	0.4372E-09
79	95	109	117	0.1332E-05	75	87	109	127	0.4594E-09
75	87	109	125	0.1351E-05	75	79	117	123	0.4692E-09
77	91	117	127	0.1352E-05	77	91	117	127	0.4700E-09

BEST AND WORST CODES-4 AND 6 DB-BIT AND BURST ERRORS-IN ORDER

75	95	109	121	0.3134E-05	0.1583E-05	0.7798E-06	0.3742E-06	0.1748E-06
		0.7932E-07		0.3494E-07	0.1491E-07	0.6159E-08	0.2456E-08	0.9440E-09
75	115	117	119	0.8800E-05	0.4618E-05	0.2361E-05	0.1175E-05	0.5682E-06
		0.2667E-06		0.1212E-06	0.5330E-07	0.2262E-07	0.9255E-08	0.3641E-08
79	87	109	121	0.3251E-05	0.1629E-05	0.7946E-06	0.3768E-06	0.1736E-06
		0.7755E-07		0.3357E-07	0.1405E-07	0.5685E-08	0.2217E-08	0.8327E-09
75	115	117	119	0.8800E-05	0.4618E-05	0.2361E-05	0.1175E-05	0.5682E-06
		0.2667E-06		0.1212E-06	0.5330E-07	0.2262E-07	0.9255E-08	0.3641E-08
79	87	109	121	0.1240E-05	0.6375E-06	0.3192E-06	0.1555E-06	0.7354E-07
		0.3374E-07		0.1500E-07	0.6446E-08	0.2675E-08	0.1070E-08	0.4119E-09
75	115	117	119	0.2442E-05	0.1292E-05	0.6655E-06	0.3335E-06	0.1624E-06
		0.7666E-07		0.3505E-07	0.1549E-07	0.6607E-08	0.2714E-08	0.1072E-08
79	87	109	121	0.1240E-05	0.6375E-06	0.3192E-06	0.1555E-06	0.7354E-07
		0.3374E-07		0.1500E-07	0.6446E-08	0.2675E-08	0.1070E-08	0.4119E-09
75	115	117	119	0.2442E-05	0.1292E-05	0.6655E-06	0.3335E-06	0.1624E-06
		0.7666E-07		0.3505E-07	0.1549E-07	0.6607E-08	0.2714E-08	0.1072E-08

3

Table VI-2 (concl)

BEST TEN UNION BOUNDS AND A CODE HAVING THESE BOUNDS FOR....
V= 4 K= 8 MESSAGE VECTORS=1-1111111111 MAXIMUM-MINIMUM WEIGHT=22

4 AND 6 DB TABLES FOR BIT ERRORS

137	191	221	235	0.8407E-06	137	191	221	235	0.1197E-09
137	187	215	253	0.8608E-06	137	187	215	253	0.1202E-09
137	173	223	239	0.8857E-06	137	173	223	239	0.1251E-09
137	173	239	251	0.9114E-06	137	181	247	251	0.1262E-09
137	181	247	251	0.9140E-06	137	173	239	251	0.1267E-09
137	187	239	245	0.9209E-06	137	181	223	247	0.1282E-09
137	175	237	251	0.9288E-06	137	183	223	245	0.1307E-09
137	175	237	247	0.9292E-06	137	175	237	251	0.1318E-09
137	183	223	245	0.9332E-06	137	175	223	237	0.1320E-09
137	175	223	237	0.9358E-06	137	187	239	245	0.1324E-09

4 AND 6 DB TABLES FOR BURST ERRORS

137	191	221	235	0.3533E-06	137	191	221	235	0.6387E-10
137	187	215	253	0.3563E-06	137	187	215	253	0.6396E-10
137	175	221	251	0.3651E-06	139	155	239	245	0.6527E-10
137	215	221	251	0.3687E-06	137	215	221	251	0.6563E-10
137	187	223	245	0.3698E-06	137	175	221	251	0.6570E-10
137	173	223	239	0.3708E-06	137	173	223	239	0.6586E-10
137	187	239	245	0.3725E-06	137	181	247	251	0.6593E-10
137	187	223	235	0.3732E-06	137	187	223	235	0.6595E-10
137	181	247	251	0.3734E-06	137	187	223	245	0.6600E-10
137	173	239	251	0.3738E-06	137	173	239	251	0.6614E-10

BEST AND WORST CODES-4 AND 6 DB-BIT AND BURST ERRORS-IN ORDER

137	191	221	235	0.8407E-06	0.3996E-06	0.1847E-06	0.8291E-07	0.3611E-07
		0.1524E-07		0.6220E-08	0.2452E-08	0.9319E-09	0.3408E-09	0.1197E-09
139	155	183	191	0.2301E-05	0.1125E-05	0.5346E-06	0.2465E-06	0.1101E-06
		0.4760E-07		0.1987E-07	0.7995E-08	0.3096E-08	0.1151E-08	0.4102E-09
137	191	221	235	0.8407E-06	0.3996E-06	0.1847E-06	0.8291E-0.	0.3611E-07
		0.1524E-07		0.6220E-08	0.2452E-08	0.9319E-09	0.3408E-09	0.1197E-09
139	155	189	191	0.2188E-05	0.1075E-05	0.5131E-06	0.2379E-06	0.1069E-06
		0.4647E-07		0.1952E-07	0.7907E-08	0.3083E-08	0.1154E-08	0.4143E-09
137	191	221	235	0.3533E-06	0.1727E-06	0.8210E-07	0.3788E-07	0.1694E-07
		0.7329E-08		0.3064E-08	0.1235E-08	0.4794E-09	0.1787E-09	0.6387E-10
139	155	183	191	0.7265E-06	0.3602E-06	0.1735E-06	0.8108E-07	0.3669E-07
		0.1605E-07		0.6779E-08	0.2758E-08	0.1079E-08	0.4053E-09	0.1458E-09
137	191	221	235	0.3533E-06	0.1727E-06	0.8210E-07	0.3788E-07	0.1694E-07
		0.7329E-08		0.3064E-08	0.1235E-08	0.4794E-09	0.1787E-09	0.6387E-10
139	155	183	191	0.7265E-06	0.3602E-06	0.1735E-06	0.8108E-07	0.3669E-07
		0.1605E-07		0.6779E-08	0.2758E-08	0.1079E-08	0.4053E-09	0.1458E-09

Table VI-3 Tabulation of Good Codes

GOOD CODES FOR....V= 2 K= 4 MESSAGE VECTORS=1-1111111111 MAXIMUM-MINIMUM WEIGHT= 6

REV. CON. VEC. #1	REV. CON. VEC. #2	REV. CON. VEC. #3	REV. CON. VEC. #4	CON. VEC. #1	CON. VEC. #2	CON. VEC. #3	CON. VEC. #4	CODE WORD WEIGHTS FOR MAX-MIN	+1	+2	UNION BOUNDS 4DB	6DB	RECORD NUMBER
11	13			11	13			4		38	0.4072E-03	0.2393E-05	1
11	13			11	13			2		10	0.1521E-03	0.1112E-05	1
13	15			11	15			2	7	18	0.3505E-03	0.1699E-05	2
13	15			11	15			1	3	5	0.1294E-03	0.7634E-06	2

GOOD CODES FOR....V= 2 K= 5 MESSAGE VECTORS=1-1111111111 MAXIMUM-MINIMUM WEIGHT= 7

REV. CON. VEC. #1	REV. CON. VEC. #2	REV. CON. VEC. #3	REV. CON. VEC. #4	CON. VEC. #1	CON. VEC. #2	CON. VEC. #3	CON. VEC. #4	CODE WORD WEIGHTS FOR MAX-MIN	+1	+2	UNION BOUNDS 4DB	6DB	RECORD NUMBER
25	29			19	23			7	10	13	0.1783E-03	0.5661E-06	1
25	29			19	23			3	3	3	0.6257E-04	0.2264E-06	1
25	27			19	27			4	12	26	0.1543E-03	0.4002E-06	2
25	27			19	27			2	4	6	0.5411E-04	0.1723E-06	2
23	25			19	29			4	12	20	0.1455E-03	0.3929E-06	3
23	25			19	29			2	3	4	0.4834E-04	0.1618E-06	3

Table VI-3 (cont)

GOOD CODES FOR....V= 2   K= 8   MESSAGE VECTORS=1-1111111111   MAXIMUM-MINIMUM WEIGHT=10

REV. CON. VEC. #1	REV. CON. VEC. #2	REV. CON. VEC. #3	REV. CON. VEC. #4	CON. VEC. #1	CON. VEC. #2	CON. VEC. #3	CON. VEC. #4	CODE WORD WEIGHTS FOR MAX-MIN	+1	+2	UNION BOUNDS 4DB	UNION BOUNDS 6DB	RECORD NUMBER
161	247			133	239			9		70	0.4225E-05	0.1438E-08	1
161	247			133	239			4		16	0.1476E-05	0.6001E-09	1
161	239			133	247			9		64	0.4094E-05	0.1423E-08	2
161	239			133	247			4		15	0.1454E-05	0.5976E-09	2
145	253			137	191			15		55	0.5524E-05	0.2241E-08	3
145	253			137	191			6		13	0.1959E-05	0.8728E-09	3
145	251			137	223			8		40	0.3392E-05	0.1227E-08	4
145	251			137	223			4		11	0.1386E-05	0.5884E-09	4
145	247			137	239			14		63	0.5366E-05	0.2119E-08	5
145	247			137	239			5		16	0.1733E-05	0.7396E-09	5
145	239			137	247			11		76	0.4839E-05	0.1731E-08	6
145	239			137	247			4		18	0.1510E-05	0.6047E-09	6
145	223			137	251			12		53	0.4720E-05	0.1817E-08	7
145	223			137	251			5		13	0.1693E-05	0.7330E-09	7
145	191			137	253			20		50	0.6823E-05	0.2930E-08	8
145	191			137	253			7		12	0.2217E-05	0.1011E-08	8
209	253			139	191			8	22	28	0.5042E-05	0.1622E-08	9
209	253			139	191			3	6	6	0.1520E-05	0.5509E-09	9
209	247			139	239			6	29	42	0.5216E-05	0.1499E-08	10
209	247			139	239			2	7	9	0.1363E-05	0.4354E-09	10
209	239			139	247			14	17	32	0.6418E-05	0.2383E-08	11
209	239			139	247			4	5	7	0.1731E-05	0.6748E-09	11
177	253			141	191			8	19	22	0.4679E-05	0.1552E-08	13
177	253			141	191			3	5	5	0.1422E-05	0.5301E-09	13
177	247			141	239			6	19	30	0.4237E-05	0.1288E-08	14
177	247			141	239			2	5	7	0.1182E-05	0.3944E-09	14
177	239			141	247			16	26	30	0.7536E-05	0.2820E-08	15
177	239			141	247			4	6	7	0.1804E-05	0.6931E-09	15
233	241			143	151			21		99	0.8131E-05	0.3188E-08	18
233	241			143	151			7		22	0.2420E-05	0.1035E-08	18
217	241			143	155			31		82	0.1042E-04	0.4545E-08	19
217	241			143	155			8		17	0.2585E-05	0.1162E-08	19
185	241			143	157			25		80	0.8776E-05	0.3701E-08	20
185	241			143	157			7		18	0.2339E-05	0.1025E-08	20
213	241			143	171			7		119	0.4627E-05	0.1274E-08	21
213	241			143	171			3		28	0.1439E-05	0.4886E-09	21
181	241			143	173			16		89	0.6499E-05	0.2463E-08	22
181	241			143	173			5		21	0.1852E-05	0.7521E-09	22
205	241			143	179			30		116	0.1077E-04	0.4485E-08	23
205	241			143	179			7		24	0.2438E-05	0.1039E-08	23
173	241			143	181			11		63	0.4698E-05	0.1703E-08	24

Table VI-3 (cont)

GOOD CODES FOR....V= 2 K= 8 MESSAGE VECTORS=1-1111111111 MAXIMUM-MINIMUM WEIGHT=10

REV. CON. VEC. #1	REV. CON. VEC. #2	REV. CON. VEC. #3	REV. CON. VEC. #4	CON. VEC. #1	CON. VEC. #2	CON. VEC. #3	CON. VEC. #4	CODE WORD WEIGHTS FOR MAX-MIN	+1	+2	UNICN BOUNDS 4DB	6DB	RECORD NUMBER
173	241			143	181			4		15	0.1472E-05	0.5981E-09	24
157	241			143	185			10		62	0.4402E-05	0.1560E-08	25
157	241			143	185			4		15	0.1474E-05	0.5982E-09	25
221	241			143	187			6	29	32	0.5198E-05	0.1484E-08	26
221	241			143	187			2	7	7	0.1359E-05	0.4322E-09	26
189	241			143	189			14	10	26	0.5634E-05	0.2232E-08	27
189	241			143	189			4	4	6	0.1621E-05	0.6532E-09	27
211	241			143	203			10		97	0.5021E-05	0.1642E-08	28
211	241			143	203			4		21	0.1576E-05	0.6120E-09	28
179	241			143	205			7		76	0.3794E-05	0.1172E-08	29
179	241			143	205			3		19	0.1269E-05	0.4674E-09	29
203	241			143	211			27		98	0.9566E-05	0.4021E-08	30
203	241			143	211			7		19	0.2331E-05	0.1027E-08	30
171	241			143	213			9		78	0.4335E-05	0.1455E-08	31
171	241			143	213			3		20	0.1277E-05	0.4695E-09	31
155	241			143	217			10		62	0.4313E-05	0.1558E-08	32
155	241			143	217			4		14	0.1439E-05	0.5953E-09	32
187	241			143	221			6	10	26	0.3521E-05	0.1114E-08	33
187	241			143	221			2	4	6	0.1090E-05	0.3736E-09	33
167	241			143	229			6		88	0.3794E-05	0.1061E-08	34
167	241			143	229			3		19	0.1269E-05	0.4674E-09	34
151	241			143	233			13		62	0.5128E-05	0.1978E-08	35
151	241			143	233			5		14	0.1705E-05	0.7352E-09	35
183	241			143	237			6	18	36	0.4173E-05	0.1278E-08	36
183	241			143	237			2	6	8	0.1245E-05	0.4134E-08	36
201	239			147	247			2	25	48	0.4010E-05	0.8837E-09	37
201	239			147	247			1	7	10	0.1112E-05	0.2978E-09	37
191	201			147	253			4	21	34	0.3931E-05	0.1055E-08	39
191	201			147	253			2	5	8	0.1200E-05	0.3969E-09	39
169	247			149	239			6	22	26	0.4415E-05	0.1334E-08	40
169	247			149	239			2	6	6	0.1237E-05	0.4101E-09	40
169	239			149	247			6	17	20	0.3960E-05	0.1231E-08	41
169	239			149	247			2	5	5	0.1150E-05	0.3900E-09	41
169	223			149	251			2	16	44	0.3259E-05	0.7105E-09	42
169	223			149	251			1	4	11	0.9247E-06	0.2464E-09	42
185	233			151	157			32		85	0.1069E-04	0.4690E-08	43
185	233			151	157			9		19	0.2878E-05	0.1307E-08	43
173	233			151	181			9		81	0.4455E-05	0.1464E-08	44
173	233			151	181			4		19	0.1542E-05	0.6074E-09	44
227	233			151	199			14		111	0.6449E-05	0.2237E-08	45
227	233			151	199			5		22	0.1875E-05	0.7546E-09	45

Table VI-3 (cont)

GOOD CODES FOR....V= 2 K= 8 MESSAGE VECTORS=1-1111111111 MAXIMUM-MINIMUM WEIGHT=10

REV. CON. VEC. #1	REV. CON. VEC. #2	REV. CON. VEC. #3	REV. CON. VEC. #4	CON. VEC. #1	CON. VEC. #2	CON. VEC. #3	CON. VEC. #4	CODE WORD WEIGHTS FOR MAX-MIN	+1	+2	UNION BOUNDS 4DB	6DB	RECORD NUMBER
203	233			151	211			8		97	0.4440E-05	0.1361E-08	46
203	233			151	211			3		22	0.1314E-05	0.4742E-09	46
171	233			151	213			16		49	0.5703E-05	0.2367E-08	47
171	233			151	213			5		13	0.1690E-05	0.7329E-09	47
183	233			151	237			12	10	58	0.5708E-05	0.2028E-08	48
183	233			151	237			2	4	14	0.1219E-05	0.3914E-09	48
191	233			151	253			9		73	0.4315E-05	0.1446E-08	49
191	233			151	253			3		18	0.1249E-05	0.4650E-09	49
153	253			153	191			16	45	30	0.8631E-05	0.3150E-08	50
153	253			153	191			4	11	8	0.2124E-05	0.7830E-09	50
185	217			155	157			32		104	0.1115E-04	0.4738E-08	52
185	217			155	157			8		23	0.2704E-05	0.1177E-08	52
217	229			155	167			15		104	0.6588E-05	0.2360E-08	53
217	229			155	167			4		23	0.1627E-05	0.6171E-09	53
213	217			155	171			35		108	0.1207E-04	0.5168E-08	54
213	217			155	171			8		23	0.2707E-05	0.1177E-08	54
157	217			155	185			10		94	0.5007E-05	0.1636E-08	57
157	217			155	185			4		22	0.1604E-05	0.6146E-09	57
189	217			155	189			20	17	56	0.8340E-05	0.3271E-08	58
189	217			155	189			5	5	11	0.2051E-05	0.8228E-09	58
211	217			155	203			18		121	0.7712E-05	0.2820E-08	60
211	217			155	203			6		22	0.2145E-05	0.8945E-09	60
179	217			155	205			14		99	0.6009E-05	0.2203E-08	61
179	217			155	205			5		21	0.1822E-05	0.7513E-09	61
217	243			155	207			16	17	30	0.6824E-05	0.2653E-08	62
217	243			155	207			4	5	6	0.1702E-05	0.6720E-09	62
203	217			155	211			23		98	0.8584E-05	0.3464E-08	63
203	217			155	211			7		18	0.2330E-05	0.1025E-08	63
171	217			155	213			22		45	0.7265E-05	0.3198E-08	64
171	217			155	213			6		12	0.1941E-05	0.8705E-09	64
217	251			155	223			15		65	0.5788E-05	0.2266E-08	65
217	251			155	223			5		14	0.1717E-05	0.7355E-09	65
199	217			155	227			24		115	0.9200E-05	0.3645E-08	66
199	217			155	227			6		21	0.2120E-05	0.8920E-09	66
167	217			155	229			10		75	0.4640E-05	0.1591E-08	67
167	217			155	229			3		17	0.1232E-05	0.4627E-09	67
217	247			155	239			23		94	0.8498E-05	0.3454E-08	68
217	247			155	239			5		20	0.1829E-05	0.7496E-09	68
175	217			155	245			6	15	26	0.3870E-05	0.1204E-08	69
175	217			155	245			1	5	8	0.9286E-06	0.2564E-09	69
217	239			155	247			29		89	0.9999E-05	0.4281E-08	70

Table VI-3 (cont)

GOOD CODES FOR....V= 2 K= 8 MESSAGE VECTORS=1-1111111111 MAXIMUM-MINIMUM WEIGHT=10

REV. CON. VEC. #1	REV. CON. VEC. #2	REV. CON. VEC. #3	REV. CON. VEC. #4	CON. VEC. #1	CON. VEC. #2	CON. VEC. #3	CON. VEC. #4	CODE WORD WEIGHTS FOR MAX-MIN	+1	+2	UNION BOUNDS 4DB	6DB	RECORD NUMBER
217	239			155	247			7		20	0.2361E-05	0.1029E-08	70
159	217			155	249			10	12	20	0.4633E-05	0.1695E-08	71
159	217			155	249			3	4	5	0.1346E-05	0.5113E-09	71
217	223			155	251			44		103	0.1437E-04	0.6415E-08	72
217	223			155	251			10		20	0.3182E-05	0.1449E-08	72
191	217			155	253			9		75	0.4368E-05	0.1451E-08	73
191	217			155	253			3		18	0.1254E-05	0.4651E-09	73
185	213			157	171			13		98	0.5842E-05	0.2064E-08	74
185	213			157	171			4		21	0.1576E-05	0.6120E-09	74
181	185			157	173			23		104	0.8724E-05	0.3479E-08	75
181	185			157	173			7		21	0.2397E-05	0.1032E-08	75
173	185			157	181			23		107	0.1013E-04	0.4186E-08	76
173	185			157	181			8		22	0.2687E-05	0.1174E-08	76
185	237			157	183			8	32	28	0.5670E-05	0.1798E-08	77
185	237			157	183			2	8	8	0.1410E-05	0.4507E-09	77
157	185			157	185			27		102	0.9696E-05	0.4032E-08	78
157	185			157	185			7		23	0.2425E-05	0.1037E-08	78
185	253			157	191			11		88	0.5153E-05	0.1761E-08	79
185	253			157	191			4		20	0.1565E-05	0.6099E-09	79
185	227			157	199			19		57	0.6673E-05	0.2806E-08	80
185	227			157	199			6		12	0.1941E-05	0.8705E-09	80
185	211			157	203			5		116	0.4041E-05	0.9872E-09	81
185	211			157	203			2		26	0.1128E-05	0.3439E-09	81
185	251			157	223			16		57	0.5849E-05	0.2386E-08	82
185	251			157	223			5		15	0.1725E-05	0.7376E-09	82
185	199			157	227			10		67	0.4501E-05	0.1572E-08	83
185	199			157	227			4		13	0.1431E-05	0.5933E-09	83
185	223			157	251			9		73	0.4312E-05	0.1445E-08	84
185	223			157	251			3		18	0.1251E-05	0.4650E-09	84
205	249			159	179			10	49	60	0.8159E-05	0.2470E-08	86
205	249			159	179			3	11	11	0.1968E-05	0.6533E-09	86
211	249			159	203			12	10	22	0.5086E-05	0.1946E-08	87
211	249			159	203			3	4	5	0.1334E-05	0.5110E-09	87
179	249			159	205			6	25	36	0.4560E-05	0.1399E-08	88
179	249			159	205			2	7	8	0.1302E-05	0.4307E-09	88
167	249			159	229			2	22	40	0.3643E-05	0.8113E-09	89
167	249			159	229			1	6	9	0.1029E-05	0.2779E-09	89
191	197			163	253			8	18	24	0.4816E-05	0.1548E-08	90
191	197			163	253			3	4	5	0.1371E-05	0.5131E-09	90
213	229			167	171			30		92	0.1043E-04	0.4431E-08	91
213	229			167	171			8		18	0.2614E-05	0.1165E-08	91

Table VI-3 (cont)

GOOD CODES FOR....V= 2 K= 8 MESSAGE VECTORS=1-1111111111 MAXIMUM-MINIMUM WEIGHT=10

REV. CON. VEC. #1	REV. CON. VEC. #2	REV. CON. VEC. #3	REV. CON. VEC. #4	CON. VEC. #1	CON. VEC. #2	CON. VEC. #3	CON. VEC. #4	CODE WORD WEIGHTS FOR MAX-MIN	+1	+2	UNION BOUNDS 4DB	6DB	RECORD NUMBER
181	229			167	173			38		106	0.1277E-04	0.5581E-08	92
181	229			167	173			9		22	0.2946E-05	0.1314E-08	92
173	229			167	181			15		97	0.6408E-05	0.2342E-08	93
173	229			167	181			5		20	0.1830E-05	0.7497E-09	93
189	229			167	189			16	20	38	0.7178E-05	0.2725E-09	94
189	229			167	189			4	6	8	0.1805E-05	0.6943E-09	94
229	253			167	191			34		88	0.1143E-04	0.4982E-08	95
229	253			167	191			9		19	0.2898E-05	0.1307E-08	95
211	229			167	203			30		77	0.1009E-04	0.4394E-08	97
211	229			167	203			7		14	0.2254E-05	0.1015E-08	97
179	229			167	205			12		100	0.5588E-05	0.1928E-08	98
179	229			167	205			4		19	0.1527E-05	0.6070E-09	98
229	243			167	207			12	28	40	0.6658E-05	0.2311E-08	99
229	243			167	207			3	8	8	0.1663E-05	0.5897E-09	99
203	229			167	211			12		69	0.5102E-05	0.1857E-08	100
203	229			167	211			4		15	0.1474E-05	0.5982E-09	100
171	229			167	213			19		53	0.6597E-05	0.2797E-08	101
171	229			167	213			6		11	0.1918E-05	0.8680E-09	101
229	251			167	223			17		127	0.7577E-05	0.2695E-08	102
229	251			167	223			4		28	0.1720E-05	0.6289E-09	102
199	229			167	227			11		69	0.4788E-05	0.1716E-08	103
199	229			167	227			4		15	0.1462E-05	0.5978E-09	103
167	229			167	229			10		78	0.4696E-05	0.1598E-08	104
167	229			167	229			4		18	0.1519E-05	0.6049E-09	104
229	247			167	239			19		90	0.7372E-05	0.2886E-08	105
229	247			167	239			5		20	0.1830E-05	0.7497E-09	105
207	229			167	243			16	29	34	0.7785E-05	0.2880E-08	106
207	229			167	243			4	7	6	0.1838E-05	0.7078E-09	106
191	229			167	253			23		62	0.7890E-05	0.3379E-08	108
191	229			167	253			6		13	0.1964E-05	0.8730E-09	108
213	227			171	199			12		123	0.6110E-05	0.1984E-08	109
213	227			171	199			4		22	0.1599E-05	0.6145E-09	109
213	251			171	223			19		51	0.6608E-05	0.2793E-08	110
213	251			171	223			5		13	0.1694E-05	0.7330E-09	110
199	213			171	227			17		105	0.7126E-05	0.2642E-08	111
199	213			171	227			5		20	0.1833E-05	0.7498E-09	111
213	247			171	239			12		79	0.5237E-05	0.1879E-08	112
213	247			171	239			4		17	0.1500E-05	0.6026E-09	112
213	239			171	247			23		99	0.8565E-05	0.3465E-08	113
213	239			171	247			6		20	0.2091E-05	0.8893E-09	113
213	223			171	251			28		93	0.9906E-05	0.4154E-08	114

Table VI-3 (cont)

GOOD CODES FOR....V= 2   K= 8   MESSAGE VECTORS=1-1111111111   MAXIMUM-MINIMUM WEIGHT=10

REV. CON. VEC. #1	REV. CON. VEC. #2	REV. CON. VEC. #3	REV. CON. VEC. #4	CON. VEC. #1	CON. VEC. #2	CON. VEC. #3	CON. VEC. #4	CODE WORD WEIGHTS FOR MAX-MIN	+1	+2	UNION BOUNDS 4DB	6DB	RECORD NUMBER
213	223			171	251			7		20	0.2376E-05	0.1030E-08	114
181	205			173	179			22		65	0.7690E-05	0.3246E-08	115
181	205			173	179			6		15	0.2006E-05	0.8778E-09	115
181	211			173	203			17		86	0.6710E-05	0.2595E-08	116
181	211			173	203			5		17	0.1764E-05	0.7423E-09	116
181	203			173	211			13		77	0.5459E-05	0.2014E-08	117
181	203			173	211			4		16	0.1476E-05	0.6001E-09	117
181	251			173	223			10		58	0.4331E-05	0.1551E-08	118
181	251			173	223			3		14	0.1178E-05	0.4557E-09	118
181	231			173	231			6	35	52	0.5760E-05	0.1628E-08	119
181	231			173	231			1	9	12	0.1267E-05	0.3374E-09	119
181	247			173	239			9		67	0.4196E-05	0.1431E-08	120
181	247			173	239			3		16	0.1214E-05	0.4603E-09	120
181	207			173	243			4	10	52	0.3584E-05	0.9019E-09	121
181	207			173	243			1	4	11	0.9314E-06	0.2464E-09	121
205	253			179	191			41		97	0.1349E-04	0.5982E-08	124
205	253			179	191			9		20	0.2918E-05	0.1310E-08	124
205	227			179	199			26		65	0.8731E-05	0.3805E-08	125
205	227			179	199			7		12	0.2208E-05	0.1010E-08	125
199	205			179	227			28		118	0.1037E-04	0.4213E-08	126
199	205			179	227			7		19	0.2357E-05	0.1027E-08	126
205	247			179	239			29		85	0.9953E-05	0.4273E-08	127
205	247			179	239			7		16	0.2288E-05	0.1020E-08	127
205	239			179	247			16		75	0.6278E-05	0.2431E-08	128
205	239			179	247			5		15	0.1736E-05	0.7379E-09	128
191	205			179	253			10		56	0.4319E-05	0.1547E-08	129
191	205			179	253			3		14	0.1184E-05	0.4558E-09	129
191	237			183	253			4	13	42	0.3598E-05	0.9342E-09	130
191	237			183	253			2	3	9	0.1086E-05	0.3636E-09	130
191	221			187	253			8	11	28	0.4171E-05	0.1417E-08	131
191	221			187	253			3	3	6	0.1283E-05	0.4952E-09	131
189	211			189	203			6	22	42	0.4647E-05	0.1367E-08	132
189	211			189	203			1	6	11	0.1036E-05	0.2805E-09	132
189	247			189	239			8	19	36	0.4798E-05	0.1576E-08	133
189	247			189	239			3	5	7	0.1425E-05	0.5329E-09	133
211	253			191	203			9		86	0.4631E-05	0.1478E-08	134
211	253			191	203			3		19	0.1275E-05	0.4676E-09	134
203	253			191	211			14		104	0.[illegible]324E-05	0.2220E-08	135
203	253			191	211			4		22	0.1602E-05	0.6146E-09	135
203	227			199	211			22		111	0.8589E-05	0.3356E-08	138
203	227			199	211			6		20	0.2095E-05	0.8895E-09	138

Table VI-3 (cont)

GOOD CODES FOR....V= 2 K= 8 MESSAGE VECTORS=1-1111111111 MAXIMUM-MINIMUM WEIGHT=10

REV. CON. VEC. #1	REV. CON. VEC. #2	REV. CON. VEC. #3	REV. CON. VEC. #4	CON. VEC. #1	CON. VEC. #2	CON. VEC. #3	CON. VEC. #4	CODE WORD WEIGHTS FOR MAX-MIN	+1	+2	UNION BOUNDS 4DB	6DB	RECORD NUMBER
211	251			203	223			28		108	0.1019E-04	0.4189E-08	139
211	251			203	223			6		22	0.2139E-05	0.8943E-09	139
211	247			203	239			22		64	0.7751E-05	0.3246E-08	140
211	247			203	239			6		13	0.1975E-05	0.8733E-09	140
211	239			203	247			39		93	0.1283E-04	0.5692E-08	141
211	239			203	247			[illegible]		18	0.2867E-05	0.1305E-08	141
211	223			203	251			45		140	0.1537E-04	0.6643E-08	142
211	223			203	251			9		28	0.3059E-05	0.1328E-08	142

Table VI-3 (cont)

GCOD CODES FOR....V= 2   K= 9   MESSAGE VECTORS=1-1111111111   MAXIMUM-MINIMUM WEIGHT=12

REV. CON. VEC. #1	REV. CON. VEC. #2	REV. CON. VEC. #3	REV. CON. VEC. #4	CON. VEC. #1	CON. VEC. #2	CON. VEC. #3	CON. VEC. #4	CODE WORD WEIGHTS FOR MAX-MIN	+1	+2	UNION BOUNDS 4DB	6DB	RECORD NUMBER
369	493			285	367			42		154	0.1128E-05	0.1072E-09	1
369	493			285	367			12		34	0.3020E-06	0.3019E-10	1
369	477			285	375			36		172	0.1035E-05	0.9360E-10	2
369	477			285	375			12		33	0.3008E-06	0.3015E-10	2
369	491			285	431			33		179	0.9773E-06	0.8667E-10	3
369	491			285	431			11		35	0.2824E-06	0.2783E-10	3
367	425			299	493			40		161	0.1096E-05	0.1027E-09	4
367	425			299	493			11		34	0.2813E-06	0.2779E-10	4
319	425			299	505			55		141	0.1362E-05	0.1377E-09	5
319	425			299	505			15		27	0.3508E-06	0.3707E-10	5
361	463			301	487			42		172	0.1152E-05	0.1079E-09	6
361	463			301	487			11		36	0.2840E-06	0.2787E-10	6
361	415			301	499			52		139	0.1306E-05	0.1305E-09	7
361	415			301	499			12		32	0.2986E-06	0.3010E-10	7
367	489			303	493			40	24	86	0.1136E-05	0.1076E-09	8
367	489			303	493			10	8	16	0.2824E-06	0.2725E-10	8
367	409			307	493			38		145	0.1035E-05	0.9728E-10	9
367	409			307	493			11		30	0.2757E-06	0.2763E-10	9
345	477			309	375			54		150	0.1362E-05	0.1358E-09	10
345	477			309	375			15		30	0.3564E-06	0.3720E-10	10
345	431			309	491			46		140	0.1186E-05	0.1162E-09	11
345	431			309	491			13		29	0.3140E-06	0.3237E-10	11
319	345			309	505			43		149	0.1142E-05	0.1094E-09	12
319	345			309	505			12		33	0.3010E-06	0.3015E-10	12
313	443			313	443			44		131	0.1133E-05	0.1110E-09	13
313	443			313	443			13		28	0.3134E-06	0.3233E-10	13
357	477			333	375			86		159	0.2010E-05	0.2127E-09	14
357	477			333	375			19		32	0.4380E-06	0.4684E-10	14
357	507			333	447			38	50	54	0.1209E-05	0.1099E-09	15
357	507			333	447			9	12	10	0.2787E-06	0.2592E-10	15
405	471			339	471			48		155	0.1253E-05	0.1216E-09	16
405	471			339	471			13		32	0.3189E-06	0.3250E-10	16
367	405			339	493			53		146	0.1333E-05	0.1332E-09	17
367	405			339	493			13		30	0.3153E-06	0.3241E-10	17
419	501			351	395			47		201	0.1298E-05	0.1211E-09	18
419	501			351	395			13		35	0.3229E-06	0.3262E-10	18
355	501			351	397			50		144	0.1272E-05	0.1259E-09	19
355	501			351	397			14		26	0.3296E-06	0.3464E-10	19
403	471			403	471			57		189	0.1486E-05	0.1446E-09	20
403	471			403	471			14		37	0.3465E-06	0.3510E-10	20

Table VI-3 (cont)

GOOD CODES FOR....V= 3   K= 4   MESSAGE VECTORS=1-1111111111   MAXIMUM-MINIMUM WEIGHT=10

REV. CON. VEC. #1	REV. CON. VEC. #2	REV. CON. VEC. #3	REV. CON. VEC. #4	CON. VEC. #1	CON. VEC. #2	CON. VEC. #3	CON. VEC. #4	CODE WORD WEIGHTS FOR MAX-MIN	CODE WORD WEIGHTS FOR +1	CODE WORD WEIGHTS FOR +2	UNION BOUNDS 4DB	UNION BOUNDS 6DB	RECORD NUMBER
11	13	15		11	13	15		6		6	0.2013E-03	0.8606E-06	1
11	13	15		11	13	15		3		2	0.8400E-04	0.4131E-06	1

GOOD CODES FOR....V= 3   K= 5   MESSAGE VECTORS=[illegible]-1111111111   MAXIMUM-MINIMUM WEIGHT=12

REV. CON. VEC. #1	REV. CON. VEC. #2	REV. CON. VEC. #3	REV. CON. VEC. #4	CON. VEC. #1	CON. VEC. #2	CON. VEC. #3	CON. VEC. #4	CODE WORD WEIGHTS FOR MAX-MIN	CODE WORD WEIGHTS FOR +1	CODE WORD WEIGHTS FOR +2	UNION BOUNDS 4DB	UNION BOUNDS 6DB	RECORD NUMBER
21	27	31		21	27	31		12		12	0.6444E-04	0.1093E-06	1
21	27	31		21	27	31		5		3	0.2305E-04	0.4393E-07	1

Table VI-3 (cont)

GOOD CODES FOR....V= 3   K= 6   MESSAGE VECTORS=1-1111111111   MAXIMUM-MINIMUM WEIGHT=13

REV. CON. VEC. #1	REV. CON. VEC. #2	REV. CON. VEC. #3	REV. CON. VEC. #4	CON. VEC. #1	CON. VEC. #2	CON. VEC. #3	CON. VEC. #4	CODE WORD WEIGHTS FOR MAX-MIN	+1	+2	UNION BOUNDS 4DB	6DB	RECORD NUMBER
41	59	61		37	47	55		4	8	14	0.1932E-04	0.1552E-07	1
41	59	61		37	47	55		2	3	4	0.7037E-05	0.6623E-08	1
41	55	61		37	47	59		4	14	13	0.2313E-04	0.1873E-07	2
41	55	61		37	47	59		2	4	3	0.7434E-05	0.7037E-08	2
41	55	59		37	55	59		4	6	23	0.2044E-04	0.1583E-07	3
41	55	59		37	55	59		2	2	5	0.6711E-05	0.6267E-08	3
41	47	59		37	55	61		4	10	16	0.2260E-04	0.1742E-07	4
41	47	59		37	55	61		2	3	4	0.7234E-05	0.6695E-08	4
41	47	55		37	59	61		4	8	28	0.2267E-04	0.1743E-07	5
41	47	55		37	59	61		2	3	6	0.7486E-05	0.6890E-08	5
53	57	59		39	43	55		7	10	13	0.2816E-04	0.2398E-07	6
53	57	59		39	43	55		3	3	3	0.8998E-05	0.8848E-08	6
47	53	57		39	43	61		1	8	26	0.1778E-04	0.1093E-07	7
47	53	57		39	43	61		1	3	6	0.6008E-05	0.4794E-08	7
45	57	61		39	45	47		7	12	11	0.2786E-04	0.2435E-07	8
45	57	61		39	45	47		3	4	3	0.9425E-05	0.9307E-08	8
45	57	59		39	45	55		4	2	14	0.1832E-04	0.1320E-07	9
45	57	59		39	45	55		2	1	4	0.6502E-05	0.5776E-08	9
45	47	57		39	45	61		4	12	11	0.2336E-04	0.1806E-07	10
45	47	57		39	45	61		2	4	3	0.7867E-05	0.7185E-08	10
43	57	61		39	47	53		4	8	19	0.2111E-04	0.1654E-07	11
43	57	61		39	47	53		2	3	5	0.7387E-05	0.6818E-08	11
43	57	59		39	53	55		7	10	16	0.2680E-04	0.2382E-07	12
43	57	59		39	53	55		3	3	4	0.8854E-05	0.8876E-08	12
43	55	57		39	53	59		7	10	5	0.2439E-04	0.2241E-07	13
43	55	57		39	53	59		3	3	1	0.8075E-05	0.8462E-08	13
45	51	61		45	47	51		4	10	11	0.2148E-04	0.1669E-07	14
45	51	61		45	47	51		2	3	3	0.7161E-05	0.6608E-08	14

Table VI-3 (cont)

GOOD CODES FOR....V= 3 K= 7 MESSAGE VECTORS=1-1111111111 MAXIMUM-MINIMUM WEIGHT=15

REV. CON. VEC. #1	REV. CON. VEC. #2	REV. CON. VEC. #3	REV. CON. VEC. #4	CON. VEC. #1	CON. VEC. #2	CON. VEC. #3	CON. VEC. #4	CODE WORD WEIGHTS FOR MAX-MIN	+1	+2	UNION BOUNDS 4DB	6DB	RECORD NUMBER
93	105	123		75	93	111		10	10	13	0.5796E-05	0.2002E-08	1
93	105	123		75	93	111		4	4	3	0.1954E-05	0.7570E-09	1
105	107	125		75	95	107		10	20	23	0.7629E-05	0.2467E-08	2
105	107	125		75	95	107		4	6	5	0.2285E-05	0.8467E-09	2
87	105	119		75	117	119		15	18	13	0.8278E-05	0.2995E-08	3
87	105	119		75	117	119		5	5	3	0.2366E-05	0.9331E-09	3
89	117	119		77	87	119		15	10	13	0.7604E-05	0.2737E-08	4
89	117	119		77	87	119		5	3	3	0.2220E-05	0.8706E-09	4
89	93	123		77	93	111		10	14	8	0.5844E-05	0.2061E-08	5
89	93	123		77	93	111		4	5	2	0.2005E-05	0.7778E-09	5
89	95	107		77	107	125		12	10	23	0.6934E-05	0.2379E-08	6
89	95	107		77	107	125		4	4	5	0.2067E-05	0.7766E-09	6
91	117	121		79	87	109		7	8	22	0.5364E-05	0.1607E-08	7
91	117	121		79	87	109		3	3	6	0.1739E-05	0.6108E-09	7
87	109	121		79	91	117		11	14	24	0.7239E-05	0.2403E-08	8
87	109	121		79	91	117		3	4	6	0.1862E-05	0.6476E-09	8
91	93	121		79	93	109		7	10	16	0.5443E-05	0.1630E-08	9
91	93	121		79	93	109		3	4	4	0.1800E-05	0.6316E-09	9
87	107	121		79	107	117		12	12	19	0.6842E-05	0.2397E-08	10
87	107	121		79	107	117		4	4	5	0.2036E-05	0.7723E-09	10
93	101	123		83	93	111		9	4	24	0.5600E-05	0.1764E-08	11
93	101	123		83	93	111		3	2	6	0.1655E-05	0.5778E-09	11
91	101	125		83	95	109		11	16	19	0.7068E-05	0.2406E-08	12
91	101	125		83	95	109		3	5	5	0.1889E-05	0.6696E-09	12
91	95	101		83	109	125		7	16	21	0.6055E-05	0.1863E-08	13
91	95	101		83	109	125		3	5	5	0.1887E-05	0.5695E-09	13
87	101	123		83	111	117		17	14	10	0.8243E-05	0.3110E-08	14
87	101	123		83	111	117		5	4	2	0.2196E-05	0.8876E-09	14
87	101	111		83	117	123		12	18	20	0.8012E-05	0.2665E-08	15
87	101	111		83	117	123		4	5	4	0.2171E-05	0.8057E-09	15
85	115	119		85	103	119		10	14	18	0.6248E-05	0.2151E-08	16
85	115	119		85	103	119		4	5	4	0.2057E-05	0.7937E-09	16
87	115	117		87	103	117		10	22	5	0.6948E-05	0.2366E-08	17
87	115	117		87	103	117		4	5	1	0.2007E-05	0.7764E-09	17

Table VI-3 (cont)

GOOD CODES FOR....V= 4 K= 4 MESSAGE VECTORS=1-1111111111 MAXIMUM-MINIMUM WEIGHT=12

REV. CON. VEC. #1	REV. CON. VEC. #2	REV. CON. VEC. #3	REV. CON. VEC. #4	CON. VEC. #1	CON. VEC. #2	CON. VEC. #3	CON. VEC. #4	CODE WORD WEIGHTS FOR MAX-MIN	+1	+2	UNION BOUNDS 4DB	6DB	RECORD NUMBER
9	11	13	15	9	11	13	15	1		2	0.1323E-03	0.7227E-06	1
9	11	13	15	9	11	13	15	1		1	0.8471E-04	0.6083E-06	1

GOOD CODES FOR....V= 4 K= 5 MESSAGE VECTORS=1-1111111111 MAXIMUM-MINIMUM WEIGHT=16

REV. CON. VEC. #1	REV. CON. VEC. #2	REV. CON. VEC. #3	REV. CON. VEC. #4	CON. VEC. #1	CON. VEC. #2	CON. VEC. #3	CON. VEC. #4	CODE WORD WEIGHTS FOR MAX-MIN	+1	+2	UNION BOUNDS 4DB	6DB	RECORD NUMBER
21	27	29	31	21	23	27	31	8		7	0.4828E-04	0.7804E-07	1
21	27	29	31	21	23	27	31	4		2	0.1953E-04	0.3640E-07	1

GOOD CODES FOR....V= 4 K= 6 MESSAGE VECTORS=1-1111111111 MAXIMUM-MINIMUM WEIGHT=18

REV. CON. VEC. #1	REV. CON. VEC. #2	REV. CON. VEC. #3	REV. CON. VEC. #4	CON. VEC. #1	CON. VEC. #2	CON. VEC. #3	CON. VEC. #4	CODE WORD WEIGHTS FOR MAX-MIN	+1	+2	UNION BOUNDS 4DB	6DB	RECORD NUMBER
41	53	59	63	37	43	55	63	5		19	0.1315E-04	0.8492E-08	1
41	53	59	63	37	43	55	63	3		5	0.5047E-05	0.4044E-08	1
41	53	55	63	37	43	59	63	5		23	0.1413E-04	0.9029E-08	2
41	53	55	63	37	43	59	63	3		6	0.5290E-05	0.4178E-08	2
41	45	59	63	37	45	55	63	5		12	0.1211E-04	0.7763E-08	3
41	45	59	63	37	45	55	63	3		4	0.5018E-05	0.3973E-08	3
41	45	55	63	37	45	59	63	5		9	0.1180E-04	0.7447E-08	4
41	45	55	63	37	45	59	63	3		3	0.4848E-05	0.3854E-08	4
41	55	59	61	37	47	55	59	9		7	0.1508E-04	0.1141E-07	5
41	55	59	61	37	47	55	59	4		2	0.5453E-05	0.4756E-08	5
41	47	59	61	37	47	55	61	5		14	0.1238E-04	0.7962E-08	6
41	47	59	61	37	47	55	61	3		4	0.4932E-05	0.3951E-08	6
41	47	55	61	37	47	59	61	5		18	0.1353E-04	0.8561E-08	7
41	47	55	61	37	47	59	61	3		5	0.5174E-05	0.4089E-08	7

Table VI-3 (concl)

GOOD CODES FOR....V= 4 K= 6 MESSAGE VECTORS=1-1111111111 MAXIMUM-MINIMUM WEIGHT=18

REV. CON. VEC. #1	REV. CON. VEC. #2	REV. CON. VEC. #3	REV. CON. VEC. #4	CON. VEC. #1	CON. VEC. #2	CON. VEC. #3	CON. VEC. #4	CODE WORD WEIGHTS FOR MAX-MIN	+1	+2	UNION BOUNDS 4DB	6DB	RECORD NUMBER
41	43	59	63	37	53	55	63	8		18	0.1631E-04	0.1170E-07	8
41	43	59	63	37	53	55	63	4		5	0.6130E-05	0.5141E-08	8
41	43	55	63	37	53	59	63	9		14	0.1640E-04	0.1218E-07	9
41	43	55	63	37	53	59	63	4		4	0.5873E-05	0.4993E-08	9
41	47	55	59	37	55	59	61	9		12	0.1644E-04	0.1212E-07	10
41	47	55	59	37	55	59	61	4		3	0.5697E-05	0.4893E-08	10
47	53	57	61	39	43	47	61	8		10	0.1415E-04	0.1060E-07	11
47	53	57	61	39	43	47	61	4		3	0.5633E-05	0.4876E-08	11
47	53	57	59	39	43	55	61	6		17	0.1488E-04	0.9657E-08	12
47	53	57	59	39	43	55	61	3		5	0.5253E-05	0.4108E-08	12
47	53	55	57	39	43	59	61	10		13	0.1778E-04	0.1337E-07	13
47	53	55	57	39	43	59	61	4		4	0.5956E-05	0.5032E-08	13
45	57	59	61	39	45	47	55	12		2	0.1747E-04	0.1414E-07	14
45	57	59	61	39	45	47	55	5		1	0.6472E-05	0.5766E-08	14
45	55	57	61	39	45	47	59	13		7	0.1936E-04	0.1577E-07	15
45	55	57	61	39	45	47	59	5		2	0.6532E-05	0.5852E-08	15
45	47	57	61	39	45	47	61	9		19	0.1713E-04	0.1279E-07	16
45	47	57	61	39	45	47	61	4		6	0.6308E-05	0.5259E-08	16
45	47	57	59	39	45	55	61	9		7	0.1564E-04	0.1154E-07	17
45	47	57	59	39	45	55	61	4		2	0.5665E-05	0.4809E-08	17
45	47	55	57	39	45	59	61	13		2	0.1715E-04	0.1489E-07	18
45	47	55	57	39	45	59	61	5		1	0.6140E-05	0.5687E-08	18
43	57	59	61	39	47	53	55	9		17	0.1755E-04	0.1278E-07	19
43	57	59	61	39	47	53	55	4		5	0.6249E-05	0.5176E-08	19
43	55	57	61	39	47	53	59	6		17	0.1417E-04	0.9459E-08	20
43	55	57	61	39	47	53	59	3		5	0.5141E-05	0.4076E-08	20
43	47	57	61	39	47	53	61	8		10	0.1389E-04	0.1057E-07	21
43	47	57	61	39	47	53	61	4		3	0.5587E-05	0.4871E-08	21
47	51	53	61	43	47	51	61	10		10	0.1718E-04	0.1303E-07	22
47	51	53	61	43	47	51	61	4		3	0.5725E-05	0.4906E-08	22
45	47	51	61	45	47	51	61	10		6	0.1604E-04	0.1241E-07	23
45	47	51	61	45	47	51	61	4		2	0.5624E-05	0.4804E-08	23